



UNIVERSITY OF
CAMBRIDGE

Neural approaches to discourse
coherence: modeling, evaluation and
application

Youmna Farag



Murray Edwards College

This dissertation is submitted in September 2020 for the degree of Doctor of Philosophy

DECLARATION

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or am concurrently submitting, for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or is being concurrently submitted, for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. This dissertation does not exceed the prescribed limit of 60,000 words.

Younna Farag
28 September, 2020

ABSTRACT

Neural approaches to discourse coherence: modeling, evaluation and application

Youmna Farag

Discourse coherence is an important aspect of text quality that refers to the way different textual units relate to each other. In this thesis, I investigate neural approaches to modeling discourse coherence. I present a multi-task neural network where the main task is to predict a document-level coherence score and the secondary task is to learn word-level syntactic features. Additionally, I examine the effect of using contextualised word representations in single-task and multi-task setups. I evaluate my models on a synthetic dataset where incoherent documents are created by shuffling the sentence order in coherent original documents. The results show the efficacy of my multi-task learning approach, particularly when enhanced with contextualised embeddings, achieving new state-of-the-art results in ranking the coherent documents higher than the incoherent ones (96.9%). Furthermore, I apply my approach to the realistic domain of people’s everyday writing, such as emails and online posts, and further demonstrate its ability to capture various degrees of coherence.

In order to further investigate the linguistic properties captured by coherence models, I create two datasets that exhibit syntactic and semantic alterations. Evaluating different models on these datasets reveals their ability to capture syntactic perturbations but their inadequacy to detect semantic changes. I find that semantic alterations are instead captured by models that first build sentence representations from averaged word embeddings, then apply a set of linear transformations over input sentence pairs.

Finally, I present an application for coherence models in the pedagogical domain. I first demonstrate that state-of-the-art neural approaches to automated essay scoring (AES) are not robust to adversarially created, grammatical, but incoherent sequences of sentences. Accordingly, I propose a framework for integrating and jointly training a coherence model with a state-of-the-art neural AES system in order to enhance its ability to detect such adversarial input. I show that this joint framework maintains a performance comparable to the state-of-the-art AES system in predicting a holistic essay score while significantly outperforming it in adversarial detection.

ACKNOWLEDGEMENTS

As my PhD journey approaches its end, I reflect back on the fun and enjoyable times I had as well as the stressful overwhelming moments. I am grateful to the whole experience that helped me grow intellectually and personally, yet I am most grateful to the incredible people who stood by me and made this doctorate possible.

First, I would like to express my sincere gratitude to my supervisor, Ted Briscoe, for his unwavering support during my PhD. He gave me the freedom to explore my own research ideas, while guiding me along the way to stay on track. I am also deeply grateful to Helen Yannakoudakis for her much appreciated advice and support; her help was invaluable for the realisation of this work. I would like to extend my gratitude to Marek Rei and Paula Buttery for their insightful comments and suggestions on my work. Many thanks to my examiners, Andreas Vlachos and Joel Tetreault, for their valuable comments on my thesis and an intellectually stimulating and enjoyable viva.

I am also grateful to my NLP research group that created an intellectual and social environment in which we discussed research ideas, and also enjoyed the lighter moments in life. We had fun game nights and they were a lovely company in different conference trips. I'd like to thank in particular: Sian, Helen, Marek, Ahmed, Chris, Ekaterina, Andrew, Mariano, Øistein, Zheng, Russell, Paula, Menglin, Meng, Mark, Gladys, Shiva and David.

Furthermore, I would like to acknowledge my academic collaborators: Farah Nadeem for kindly testing her model on my data and Josef Valvoda for his technical help in my evaluation chapter. I extend my acknowledgement to Melek Karadag and Sami Alabed for their annotations. I would be remiss not to also thank Andrew Caines, Chris Bryant and Mariano Felice for reading my thesis and giving me useful feedback.

The Computer Lab was a welcoming environment, and in particular, I would like to thank Lise Gough for her support since I joined the Lab until the very end. I am also grateful to be part of Murray Edwards College where I met amazing people and had unforgettable memories; I would like, in particular, to thank Nicola Cavaleri who I truly appreciated her support in my first two PhD years. Finally, this doctoral research would not have taken place without funding from the Cambridge Trust, EPSRC, and Cambridge Assessment.

My family is at the center of everything and to whom I dedicate this thesis. I am

eternally grateful to my mother Omayma and father Hussein who have always encouraged me to pursue my dreams while providing me with unconditional love and support. I cannot thank them enough for all what they have done and continue to do for me. I would like to extend my gratitude to my amazing sisters: Aya, Raghdah, and Leena who have been always there for me to fall back on; my nephews and nieces: Ahmed, Adam, Ali, Nour, and Laila who have been a source of endless fun and warmth; and my brothers-in-law: Amr and Mohammed who have been true brothers to me. I am grateful to my uncle, Mohamed, whose kindness everyone felt and was moved by, and who passed away in my final year; may he rest in peace.

Not all families are biological. I have the immense privilege to have met kindred souls in Cambridge and Cairo. In Cambridge, a new family formed of incredible friends who made my experience there very special. I would like to particularly thank: Noura, Ahmed, Leen, Samuel, Melek, Moataz, Nadi, Steven, Hana, Noor, Omar, Marwa, Ayat, Amir, Chandrima, Hel, Rachel, Fatima, Aya, Ibrahim, Salma, Yara, Mohamed, Abdullah, Josh, Charoula, Ilana and Lakshmi. I truly cherished our night walks, our philosophical conversations as well as the nonsensical ones, studying together in cafes, our MCR gatherings particularly Medwards parties, and the trips we took together; all of this has helped keep me sane during my PhD. In Cairo, I would like to express my gratitude to my life-time friends: Nourhane, Reem, Aisha, Nada, Maha, Yasmin, Enas, Fatima, Ahmed, Arwa, Shinnawy, Adham, Hashish, Khaled, Maged, Hadeel, Mahmoud and Mohanad. I cannot express how grateful I am to have you all in my life. Not only have I been blessed with your friendship back in Cairo, but your love and support continued to keep me going during my PhD, even when we were thousands of miles apart.

CONTENTS

1	Introduction	15
1.1	Coherence modeling	16
1.2	Coherence evaluation	18
1.3	Coherence application	19
1.4	Thesis aims	20
1.5	Thesis structure	22
2	Background	23
2.1	Theories and frameworks of coherence	23
2.1.1	Cohesion	23
2.1.2	Coherence relations	24
2.1.3	Discourse structure theory	26
2.1.4	Centering theory	27
2.2	Traditional approaches to coherence modeling	28
2.2.1	Semantic relatedness	28
2.2.2	Probabilistic models	30
2.2.3	Entity-based methods	31
2.2.4	Coherence relation models	33
2.3	Deep learning representations	33
2.3.1	Neural encoders	33
2.3.2	Word representations	36
2.3.3	Sentence representations	39
2.4	Neural approaches to coherence modeling	40
2.4.1	Discriminative approaches	40
2.4.2	Generative approaches	43
2.5	Multi-task learning	45
2.6	Model interpretability	47
2.7	Evaluation	49

3	Approach	53
3.1	Single-task learning model	53
3.1.1	Word representation	54
3.1.2	Sentence representation	56
3.1.3	Paragraph representation	57
3.1.4	Document representation	58
3.1.5	Scoring	59
3.2	Multi-task learning model	59
3.2.1	Multi-task learning with grammatical roles	61
3.2.2	Multi-task learning with part-of-speech tags	64
3.3	Neural syntactic models	65
3.4	Summary	66
4	Experiments	67
4.1	Datasets and preprocessing	68
4.1.1	The Wall Street Journal - synthetic data	68
4.1.2	The Grammarly Corpus of Discourse Coherence - realistic data	69
4.2	Previous neural models	70
4.2.1	Local coherence	70
4.2.2	Neural EGrid	71
4.2.3	Local coherence discriminator	72
4.2.4	Paragraph sequence	73
4.3	Training and hyperparameters	73
4.4	Binary experiments	76
4.4.1	Baselines	76
4.4.2	Results	77
4.4.3	Analysis	81
4.4.3.1	Sensitivity to sentence order	81
4.4.3.2	Subject and object prediction	82
4.4.3.3	Attention visualisation	83
4.5	Realistic data experiments	88
4.5.1	Results	89
4.5.2	Analysis	91
4.5.2.1	Performance per class	91
4.5.2.2	Transfer learning	91
4.5.2.3	Ranking coherence	92
4.5.2.4	Attention visualisation	93
4.6	Summary	96

5	Evaluation of discourse coherence	99
5.1	Syntactic structure	101
5.2	Cloze Coherence Dataset	103
5.2.1	Coherent examples	103
5.2.2	Incoherent examples	105
5.2.3	Experiments	107
5.2.4	Results	109
5.3	Controlled Linguistic Alterations Dataset	112
5.3.1	Dataset	112
5.3.2	Results	115
5.4	Summary	117
6	Application of coherence models	119
6.1	Approaches to automated essay scoring	120
6.1.1	Traditional approaches	120
6.1.2	Neural approaches	120
6.2	Evaluation against adversarial input	121
6.3	Dataset and evaluation	123
6.3.1	Dataset	123
6.3.2	Evaluation	124
6.4	Neural AES models	125
6.5	Coherence models	127
6.6	Joint learning	128
6.6.1	Approach	128
6.6.2	Variations of parameter sharing	130
6.7	Experiments	131
6.8	Results	132
6.9	Summary	138
7	Conclusion	141
7.1	Summary and findings	141
7.2	Future work	144
	Bibliography	147
A	Examples from Yahoo posts	193
B	Examples from the CLAD	195
C	Pearson’s and Spearman’s correlations for the ASAP dataset	197

LIST OF ABBREVIATIONS

AES	Automated Essay Scoring
ASAP	Automated Student Assessment Prize
BCA	Bidirectional Context with Attention
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bidirectional Long Short-Term Memory
CCD	Cloze Coherence Dataset
CLAD	Controlled Linguistic Alterations Dataset
CNN	Convolutional Neural Network
DM	Discourse Markers
EGrid	Entity Grid
ELMo	Embeddings from Language Models
GCDC	Grammarly Corpus of Discourse Coherence
GR	Grammatical Role
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
JL	Joint Learning
LC	Local Coherence
LCD	Local Coherence Discriminator
LCD-L	Local Coherence Discriminator with Language modeling
LSA	Latent Semantic Analysis
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
MSE	Mean Squared Error
MTL	Multi-Task Learning
NLI	Natural Language Inference
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
PARSEQ	Paragraph Sequence
POS	Part-of-Speech
PRA	Pairwise Ranking Accuracy

QWK	Quadratic Weighted Kappa
RNN	Recurrent Neural Network
RST	Rhetorical Structure Theory
SOX	Subject, Object and Others
STL	Single-Task Learning
SVM	Support Vector Machine
TPRA	Total Pairwise Ranking Accuracy
UD	Universal Dependencies
WSD	Word Sense Disambiguation
WSJ	Wall Street Journal

INTRODUCTION

In a written discourse, the writer’s aim is to be understood and convey their intended meaning to their readers in an organised and well-formed manner. The readers, at the other end, leverage the relations between discourse segments to make the connections and inferences necessary for comprehension.¹ These relations that tie textual units together to compose a meaningful content are what we refer to as *discourse coherence*. Coherence is, therefore, a property of text that describes the way propositions are linked together to facilitate a logical flow of information and form a meaningful unified whole as the discourse unfolds. There are various aspects that contribute to discourse coherence, ranging from overt linguistic devices realised at the surface of text, such as anaphoric references and repetition of words (Halliday and Hasan, 1976; Morris and Hirst, 1991) to pragmatic relations inferred by world knowledge (Levinson, 1983; Redeker, 1990). A discourse is formed by intertwining these properties and not just relying on one type. A coherent discourse is not a set of random sentences; these sentences are rather connected to represent a certain idea/topic. For instance, the following example (a) is a coherent text from the book ‘*Outliers: The Story of Success*’ (Gladwell, 2017, p. 77), whereas (b) is an incoherent distracted text from a patient of schizophrenia (Iter et al., 2018):

- (a) “*One of the most widely used intelligence tests is something called Raven’s Progressive Matrices. It requires no language skills or specific body of acquired knowledge. It’s a measure of abstract reasoning skills. A typical Raven’s test consists of forty-eight items, each one harder than the one before it, and IQ is calculated based on how many items are answered correctly.*”
- (b) “*When I was three years old, I made my first escape attempt. I had a [unintelligible] sticker in the window. Like everybody listened to AM radio in the sixties. They had a garage band down the street. I couldn’t understand why the shoes were up on the wire. That means there was drug deal in the*

¹This also applies to spoken discourses, but I focus on written texts in the scope of this thesis.

neighborhood.”

Coherence can be described as local or global; *local coherence* refers to the relatedness between successive sentences, whereas *global coherence* looks at the structure and topic of the discourse as a whole. According to Graesser et al. (1994), local coherence is achieved when “conceptual connections relate the content of adjacent text constituents (i.e., a phrase, proposition, or clause)”, while global coherence is achieved when “most or all of the constituents can be linked together by one or more overarching themes”. Similarly, Van Dijk (1980) defines local coherence in terms of “pairwise relations between sentences of a textual sequence”, and global coherence by the notions of ‘idea’, ‘theme’, ‘gist’ and ‘upshot’ of a discourse. Local coherence is, thereby, necessary to achieve global coherence (Marcu, 1997; Barzilay and Lapata, 2005) and both levels are important to form a coherent discourse. For instance, the previous example (a) exhibits the two levels, where local coherence between successive sentences is realised by means such as referential pronouns (e.g., ‘it’ in the second and third sentences) or semantic relations (e.g., the third sentence elaborates the second.), while global coherence is achieved by focusing on one idea (i.e., Raven’s test).

This thesis contributes to research on discourse coherence from three perspectives: modeling, evaluation and application. In the remainder of this chapter, I will give an overview of these three directions and my work in relation to them, state my thesis aims and present the structure of the thesis.

1.1 Coherence modeling

Since the 1970s, various theories have been proposed to explicate what makes a discourse coherent and study the relations between discourse elements, including lexicogrammatical (Halliday and Hasan, 1976; Webber, 1988; Hoey, 2005), entity-based (Joshi and Weinstein, 1981; Gordon et al., 1993; Grosz et al., 1995), psychological (Kintsch and Van Dijk, 1978; Graesser et al., 1994; Givón, 1995), semantic (Hobbs, 1979; Redeker, 1990; Sanders et al., 1992), pragmatic (Widdowson, 1978; Van Dijk, 1979; Lascarides and Asher, 1991) and structural (Danes, 1974; Grosz and Sidner, 1986; Mann and Thompson, 1988) theories. Such studies have provided a foundational framework for *coherence modeling* which aims to estimate text coherence with computational models. For instance, lexical chains of semantically-related words occurring in consecutive sentences have been utilised as a proxy for text coherence (Morris and Hirst, 1991; Barzilay and Elhadad, 1997; Silber and McCoy, 2002; Somasundaran et al., 2014), inspired by the work of Halliday and Hasan (1976) on lexical cohesion. Other models have leveraged the notion of semantic relatedness between co-occurring words to measure the similarity between their encompassing sentences, where a higher degree of similarity between neighbouring sentences indicates a more coherent text (Foltz et al., 1998; Higgins et al., 2004; Yannakoudakis and Briscoe,

2012). Theories that describe discourse structure have also been translated into coherence assessment models. For instance, Lin et al. (2011), Feng et al. (2014) and Mesgar and Strube (2015) computationally estimated coherence by leveraging the semantic/rhetorical relations between text parts as described by Rhetorical Structure Theory (RST; Mann and Thompson, 1988), and Louis and Nenkova (2012) modeled the intentional discourse structure (Grosz and Sidner, 1986) that defines a discourse in terms of the communicative purposes of its segments. Furthermore, *Centering theory* (Grosz et al., 1995) has been the basis of a plethora of coherence models (Miltisakaki and Kukich, 2000; Karamanis, 2001; Hasler, 2004; Karamanis et al., 2004; Rus and Niraula, 2012a); it focuses on the distribution and realisation of entities across sentences, deriving from the premise that sentences should be *about* the same entities to form a coherent discourse. The *Entity Grid* (*EGrid*) model (Barzilay and Lapata, 2005, 2008) is one of the key coherence models that spurred from Centering theory; it creates an abstract representation of text that tracks entity distribution and the transition of the syntactic roles entities take across sentences. The EGrid approach has been adapted and further enhanced in numerous coherence models (Elsner et al., 2007; Filippova and Strube, 2007; Burstein et al., 2010; Cheung and Penn, 2010a; Elsner and Charniak, 2011b; Feng and Hirst, 2012; Guinaudeau and Strube, 2013).

More recently, and with the advancement of deep learning in Natural Language Processing (NLP), neural networks have been adopted in coherence modeling and outperformed traditional statistical models. A few approaches operate on structured text by incorporating EGrid representations of text as input to a neural model (Tien Nguyen and Joty, 2017; Joty et al., 2018). Other approaches are end-to-end with some focusing on capturing global context (Li and Jurafsky, 2017; Logeswaran et al., 2018; Cui et al., 2018; Bohn et al., 2019; Kumar et al., 2020a) and others focusing on capturing local coherence (Li and Hovy, 2014; Cui et al., 2017; Mesgar and Strube, 2018; Xu et al., 2019). In contrast to previous methods that focused on one aspect of coherence (e.g., lexical features, rhetorical relations or entity distribution), and in many cases relied on handcrafted features or external tools (i.e., parsers), neural end-to-end approaches take advantage of the ability of neural networks to automatically learn relevant features from unstructured text. They capture discourse-related properties by only utilising input word representations that are initialised from semantically-rich pre-trained spaces, either standard (Mikolov et al., 2013c; Zou et al., 2013; Pennington et al., 2014; Mikolov et al., 2018) or contextualised (Peters et al., 2018; Devlin et al., 2019) as will be explained in §2.3.2. I further discuss different traditional and neural approaches for discourse coherence in the next chapter.

In this thesis, I extend this line of work and propose a neural *Multi-Task Learning* (*MTL*) approach to coherence modeling. MTL has been widely leveraged in machine learning models, where a model exploits training signals from related tasks to enhance

its performance on its main task (Caruana, 1997; Ruder, 2017). My MTL model is a hierarchical neural network that learns to predict a document-level coherence score (at the network’s top layers) along with word-level syntactic information (at the bottom layers), taking advantage of inductive transfer between the two tasks. My choice of the word-level auxiliary task is inspired by previous studies that have utilised syntactic properties in coherence modeling as indicators of entity salience by using grammatical roles (GRs) (Grosz et al., 1995; Barzilay and Lapata, 2008), or the intentional structure of discourse by using Part-of-Speech (POS) tags (Louis and Nenkova, 2012), as will be detailed in Chapter 3. In contrast to neural EGrid approaches, MTL limits the use of syntactic parsers to training time as syntactic labels are learned and not fed as input features, which facilitates generalisation to new test sets.

1.2 Coherence evaluation

Barzilay and Lapata (2005, 2008) presented coherence evaluation as a binary task where a model should discriminate between coherent and incoherent documents. To that end, they proposed creating synthetic datasets, where the sentences in source coherent documents are shuffled to construct incoherent texts with the underlying assumption that the sentence order in an original document is more coherent than its permuted versions. Consequently, evaluation is carried out in a pairwise fashion, where a coherence model should be able to rank a coherent source document higher than its noisy counterparts. The news domain has become a ubiquitous source for creating coherence datasets. Barzilay and Lapata (2005, 2008) created two datasets of news articles about earthquakes and aviation accidents, while Elsner and Charniak (2008) proposed to use the Wall Street Journal (WSJ) portion of the Penn Treebank; these datasets have been widely adopted in coherence modeling research.

Leveraging synthetic data has become dominant in coherence modeling as it is easy to create and upscale. Nonetheless, there have been some attempts to create more realistic data, annotated by humans. For instance, there have been efforts in the pedagogical domain to assess coherence quality in student essays and test how strongly models agree with human graders (Higgins et al., 2004; Burstein et al., 2010; Crossley and McNamara, 2011; Burstein et al., 2013; Somasundaran et al., 2014). More recently, Lai and Tetreault (2018) released a dataset for coherence assessment of texts written by non-professional writers in everyday contexts (e.g., Yahoo posts and emails from Hillary Clinton’s office and Enron). The dataset is annotated by human judges with three degrees of coherence: low, medium and high.

In this thesis, I follow previous work and train and evaluate my coherence models on synthetic data (WSJ) (Elsner and Charniak, 2008) as well as data from realistic

domains (Lai and Tetreault, 2018). Furthermore, I extend my evaluation and develop a framework to investigate the linguistic features learned by neural coherence approaches. Predicting an overall coherence score for a document, whether in a binary domain or with multiple levels of coherence, does not tell much about what the models actually learn. This is particularly problematic in deep learning where neural networks are hard to interpret, and becomes even more challenging in a complex task like coherence assessment where many factors contribute to the coherence of a discourse (as will be elaborated in §2.1). Attempting to pinpoint the linguistic phenomena captured by neural discourse models and creating datasets that facilitate this has been a neglected area of research, which motivated me to devise datasets that exhibit syntactic and semantic alterations and examine the ability of the models to detect them.

1.3 Coherence application

Coherence is an inherent property of discourse quality and thus modeling it has various NLP applications. For example, in the domain of mental health, measuring discourse incoherence could help detect symptoms of illnesses that cause disorder in language such as schizophrenia, Alzheimer’s disease and mild strokes (Elvevåg et al., 2007; Ditman and Kuperberg, 2010; Bedi et al., 2015; Barker et al., 2017; Iyer et al., 2018; Paulino et al., 2018). Furthermore, in the pedagogical domain, evaluating coherence in student essays has gained much attention as it is an important dimension of writing competence (Mitsakaki and Kukich, 2000; Higgins and Burstein, 2007; Burstein et al., 2010; Rus and Niraula, 2012b; Yannakoudakis and Briscoe, 2012; Somasundaran et al., 2014; Feng et al., 2014; Palma and Atkinson, 2018; Tay et al., 2018; Nadeem et al., 2019). Additionally, coherence approaches have been widely employed in readability assessment since coherence is strongly associated with readability, where the more coherent a text is, the easier it is to read (Graesser et al., 2004; Crossley et al., 2007; Barzilay and Lapata, 2008; Pitler and Nenkova, 2008; Li and Hovy, 2014; Mesgar and Strube, 2015, 2016; Xia et al., 2016). This is useful in pedagogy as measuring the difficulty of reading a text helps teachers select reading comprehension tasks based on students’ abilities. In addition, coherence modeling has been frequently paired with information insertion and information ordering tasks. In information insertion, a sentence is pulled out of a text and the model is tasked with inserting it back in its original place (Chen et al., 2007; Elsner and Charniak, 2008, 2011b; Guinaudeau and Strube, 2013; Tien Nguyen and Joty, 2017); this is useful in community edited web resources such as Wikipedia that require continuous update and insertion of new information (Chen et al., 2007). In information ordering, a model is asked to organise a given set of sentences to form a coherent text (Lapata, 2003; Barzilay and Lee, 2004; Bollegala et al., 2006; Gong et al., 2016; Li and Jurafsky, 2017; Cui et al., 2018; Logeswaran

et al., 2018; Yin et al., 2019a; Wang and Wan, 2019; Oh et al., 2019; Kumar et al., 2020a), which has utility in text generation applications such as ordering the sentences produced by multi-document summarisers (Lapata, 2003). Accordingly, coherence modeling has been leveraged in summarisation tasks, either by rating the coherence of human or machine generated summaries (Barzilay and Lapata, 2008; Pitler et al., 2010; Feng and Hirst, 2012; Zhang et al., 2015; Tien Nguyen and Joty, 2017), or generating coherent summaries for documents (Barzilay and Elhadad, 1997, 2002; Barzilay and Lee, 2004; Barzilay and McKeown, 2005; Parveen and Strube, 2015; Koto et al., 2019). It has also been integrated in other text generation tasks such as machine translation (Meyer et al., 2012; Hardmeier, 2014; Smith et al., 2016b; Joty et al., 2017; Born et al., 2017; Bawden et al., 2018) and story generation (McIntyre and Lapata, 2010; Clark et al., 2018). Other coherence applications include authorship attribution (Feng and Hirst, 2014; Ferracane et al., 2017), information retrieval (Petersen et al., 2015), text segmentation (Wang et al., 2017a; Glavaš and Somasundaran, 2020), question answering (Verberne et al., 2007) and conversation thread disentanglement and reconstruction (Elsner and Charniak, 2011a; Joty et al., 2018).

This wide variety of discourse coherence applications is my main motivation to contribute to discourse coherence research. As an application to coherence modeling, I apply my coherence models to the pedagogical domain and show that integrating them to a state-of-the-art neural Automated Essay Scoring (AES) model enhances its ability to capture discourse-related features. More concretely, I demonstrate that state-of-the-art AES is not well-suited to capturing adversarial input of grammatical but incoherent sequences of sentences. To address this problem, I propose a framework for integrating and jointly training coherence models with a state-of-the-art AES model. I show that this joint learning approach can effectively capture adversarial input, further contributing to the development of an approach that strengthens AES validity.

1.4 Thesis aims

This thesis contributes to the work on discourse coherence and its main aims are as follows:

- Develop a neural model to assess text coherence; the model leverages syntactic features relevant to discourse coherence efficiently as the features are only extracted for training data. This is achieved by training the model in an MTL fashion, where the model learns to predict a document-level coherence score (as the main task) together with word-level syntactic information (as an auxiliary task), taking advantage of inductive transfer between the two tasks.
- Compare the effect of using GRs or POS tags as the labels of the auxiliary task.
- Investigate the value of initialising the model with contextualised embeddings and

whether the features learned from these embeddings are complementary to the auxiliary syntactic labels leveraged by MTL.

- Validate the MTL approach by creating other variants to the model that: perform the single task of predicting a document-level coherence score and/or incorporate the syntactic information in different fashions.
- Compare the MTL approach to state-of-the-art neural models that are either end-to-end or operate on EGrid representations of text, where the grids are required at both training and test times.
- Evaluate coherence models on the standard binary discrimination task of synthetic data where the model should rank a coherent document higher than its permuted counterparts, in addition to a stricter evaluation setting in which the model is tested on its ability to rank coherent documents higher than any incoherent/permuted document in the dataset, and not just its own permuted versions.
- Evaluate coherence models on the realistic domains of everyday writing (e.g., online posts and emails) that reflect varying degrees of coherence.
- Inspect the features the models focus on using visualisation techniques and examine quantitatively and qualitatively their biases towards certain syntactic labels.
- Create an evaluation framework for systematically investigating the syntactic and semantic features that neural coherence models learn and analysing the inter-sentential properties they capture with respect to model architecture and pre-training domain. This helps understand the models and therefore, provide insight into how to frame the task of coherence modeling and further improve the models.
- Demonstrate empirically that state-of-the-art approaches to AES are not robust against adversarially crafted essays of grammatical but incoherent sequences of sentences.
- Build a neural network that strengthens AES validity by capturing adversarial essays as well as achieving a competitive performance to state-of-the-art AES models in predicting a holistic essay score. The network jointly trains a coherence model and a neural AES system; I experiment with plugging different coherence models into the joint framework and investigate different parameter sharing setups between the coherence and AES models.

1.5 Thesis structure

The rest of the thesis is structured as follows. In Chapter 2, I put my work in context and give a background about various theories that explain discourse coherence, traditional and neural approaches to coherence modeling and different neural encoders used to generate text representations. In the same chapter, I also give an overview about MTL, highlight a few approaches used to interpret neural models and present the evaluation metrics I use. In Chapter 3, I present my MTL approach. I first describe my hierarchical model that performs the single-task of predicting a document-level coherence score then detail how it is enhanced with: auxiliary functions to predict word-level syntactic properties (GRs or POS tags) and/or contextualised word embeddings. I also discuss different approaches to incorporating syntactic information to further validate my MTL approach. Chapter 4 discusses my experimental results in coherence modeling. I present the two domains I leverage, i.e., synthetic binary data and realistic data, explain the training setup for my experiments and report the results of evaluating my models on the two domains in comparison to previous state-of-the-art approaches. Furthermore, I explicate the model performance with further analysis and visualisation techniques to understand what features the models focus on. Next, in Chapter 5, I introduce my evaluation framework for discourse models and detail the two datasets I create to better understand the models. The chapter includes results and analysis of evaluating a wide variety of neural approaches with this framework. After that in Chapter 6, I propose my joint learning framework for AES that is robust to adversarial input. I explain how a neural AES model can be integrated with different discourse models in this framework and present the results of evaluating the AES and the joint learning models on predicting holistic essay scores in addition to flagging adversarial essays. I also investigate the effect of incorporating contextualised embeddings into the evaluated models. Finally, I conclude the thesis in Chapter 7 with a summary of my work and outline possible directions for future research.

BACKGROUND

2.1 Theories and frameworks of coherence

Due to the key role coherence plays in defining a meaningful readable discourse, numerous studies have focused on investigating the features that contribute to discourse coherence. In this section, I summarise some of these theories that later formed the basis for computational models.

2.1.1 Cohesion

Cohesion is defined in terms of the lexical and grammatical devices that link text elements to one another. According to Halliday and Hasan (1976), cohesion determines whether a set of sentences has a ‘texture’ that gives it “the property of being a text”; i.e., when the interpretation of a textual unit is dependent on another. Cohesion leverages explicit linguistic cues identified at the surface of text which either connect elements in the same sentence (*intra-sentential*) or across sentences (*inter-sentential*). Halliday and Hasan (1976) classify the cohesive relations that signal coherence in text into 5 categories:

- *Reference*, which includes personal (e.g., he, she) and demonstrative (e.g., this, that) pronouns and comparatives (e.g., same, fewer). Both the referenced item and its anaphora (referencing word) refer to the same entity in the real world, example:

(1.a) Mary went shopping. *She* bought a sweater.

- *Substitution*, which occurs when an entity appearing in a sentence is substituted in the next for another that has the same structural function. The substituting item could be nominal (e.g., one, the same), verbal (e.g., do, do so) or clausal (e.g., so, not), example:

(1.b) John bought a blue sweater. Mary bought a pink *one*.

- *Ellipsis*, which occurs when an entity in a sentence is the same as a previous one and is deleted, example:

(1.c) Mary was the first person to leave the party. John was the second
 <person to leave the party>.

- *Conjunction*, which includes conjunctive phrases (i.e., discourse markers or connectives) that may be additive (e.g., furthermore, moreover), adversative (e.g., however, nevertheless), causal (e.g., therefore, thus) or temporal (e.g., afterwards, next), example:

(1.d) John studied hard for the exam. *However*, he failed.

- *Lexical cohesion*, which depends on the selection of vocabulary. It could be realised via *reiteration* by repeating the same word or using a synonym / superordinate / subordinate / general noun, or via *collocation* by using semantically related words that often co-occur, example:

(1.e) John went to the park. The *park* was empty. (reiteration by repetition)

(1.f) The *weather* is nice today. It is *sunny*. (collocation)

Reference, substitution and ellipsis can be classified as types of *grammatical cohesion*, while conjunction can be classified as a combination of both *grammatical* and *lexical* cohesion.

Although, cohesive ties are indicators of text coherence and readability (Haviland and Clark, 1974; McCulley, 1985; Haberlandt, 1982; McNamara, 2001; Duran et al., 2007; Crossley and McNamara, 2016), cohesion does not necessarily entail coherence (Carrell, 1982; Brown and Yule, 1983; Giora, 1985). I elaborate this with an example from Hobbs (1979):

(2) John took a train from Paris to Istanbul. He likes spinach.

Even though the second sentence contains a pronoun (he) that refers to an entity in the first (John), the text is not coherent. This takes us to another level of coherence achieved by semantic/pragmatic relations.

2.1.2 Coherence relations

Not all discourse relations can be expressed in terms of explicit cohesive ties and some can be defined as “the relationship between the illocutionary acts which propositions, not always overtly linked, are being used to perform.” (Widdowson, 1978, p. 28). Following Hobbs (1979), I refer to these relations as *Coherence Relations*; they are analogous to Halliday and Hasan’s 1976 ‘conjunctive relations’ but could also be implicit without recourse to

discourse connectives. They can be categorised into different types that have been widely studied in the literature, including temporal, elaborative, causal, justification, and contrast relations (Hobbs, 1979; van Dijk and Kintsch, 1983; Mann and Thompson, 1988; Hovy, 1990; Sanders et al., 1992; Lascarides and Asher, 1993; Graesser et al., 1994; Kehler and Kehler, 2002).

Coherence relations can be semantic or pragmatic. *Semantic relations* link the underlying meaning of propositions. For example, if I change (2) to “John took a train from Paris to Istanbul. He hates planes.”, it becomes more coherent because of the causality relation introduced between the two sentences. Semantic relations could be implicit or signalled by cohesive devices; examples:

(3.a) Sally is crying. Nanny has thrown out the time-worn teddy bear. (from Redeker (1990))

(3.b) Sally is crying. That is because nanny has thrown out the time-worn teddy bear.

In (3.a), we understand the causal connection between the two sentences without an explicit connective, while in (3.b) the causal connection is made explicit by leveraging the conjunctive phrase ‘That is because’. In contrast, *pragmatic relations* need world knowledge and context understanding to be inferred. Levinson (1983) defines *pragmatics* as “the study of relations between language and context that are basic to an account of language understanding”. Implicatures, for instance, are a form of pragmatics in which there is discrepancy between what is said and what is implied (Grice, 1975). Consider this example:

(4.a) There is a big party next week. Mary has to work.

It is implied that Mary will not be able to go to the party, although the text does not mention it. Other examples of pragmatics include irony:

(4.b) No one attended John’s birthday party. He is very popular.

Moreover, if it is established in example (2) that Istanbul is famous for its spinach, the example becomes more plausible. Pragmatic relations are, therefore, more challenging to capture and require better understanding of the external situational model of text. For a more detailed account of the distinction between semantic and pragmatic relations, I refer the reader to the work of Widdowson (1978), Van Dijk (1979, 1980), Schiffrin (1987), Redeker (1990) and Sanders et al. (1992).

Structure of coherence relations Due to the importance of coherence relations in forming a meaningful discourse, research efforts have been devoted to formalising how they are structured and organised in text. One of the prominent theories that describes

discourse structure is *Rhetorical Structure Theory (RST)* (Mann and Thompson, 1988), where a text is represented in a hierarchical fashion (as a tree) in which every discourse unit (tree node) is rhetorically related to other units in the text. Rhetorical relations have various types such as temporal, cause, elaboration, contrast and condition. They can also be semantic or pragmatic, and explicit (i.e., signalled by discourse connectives) or implicit. In order to facilitate the utilisation of RST, Carlson et al. (2001) released the RST Discourse Treebank (RST-DT) corpus which consists of WSJ articles annotated with rhetorical relations. RST-DT has been widely used in discourse parsing and coherence modeling.

There are other theories that formalise coherence relations such as *Discourse Lexicalized Tree Adjoining Grammar (D-LTAG)* (Webber et al., 2003) that defines relations in local contexts instead of representing the whole text as a tree. In D-LTAG, a discourse connective forms a predicate that takes two arguments (Arg1 and Arg2). Prasad et al. (2008) adopted the D-LTAG approach to annotate a portion of the WSJ and create the Penn Discourse Treebank (PDTB). Example from the PDTB is “[Third-quarter sales in Europe were exceptionally strong,]_{Arg1} boosted by promotional programs and new products – [although]_{connective} [weaker foreign currencies reduced the company’s earnings]_{Arg2}.”

2.1.3 Discourse structure theory

Grosz and Sidner (1986) describe discourse structure as three interacting components:

1. *Linguistic structure*: a discourse is divided into segments and each segment consists of a group of topically related propositions. Local coherence ties the propositions in the same segment, while global coherence exists between segments in the same discourse.
2. *Attentional structure*: at any given point in a discourse, there is a space of entities that constitute its center of attention and this space changes, according to a set of transition rules (§2.1.4), as the discourse unfolds.
3. *Intentional structure*: each proposition has a communicative goal that contributes to achieve the overall discourse purpose. Discourse intentions and their relations form the overall rationale of text.

The intentional structure, therefore, captures the purposes of the discourse segments identified by the linguistic structure, and the attentional structure abstracts the focus of attention and models how it changes throughout the discourse.

2.1.4 Centering theory

Centering theory (Grosz et al., 1995) is one of the fundamental entity-based theories that postulate the idea that a coherent discourse is ‘about’ the same entities (Chafe, 1976; Joshi and Weinstein, 1981; Prince, 1981; Grosz et al., 1983; Gordon et al., 1993).¹ The theory describes how entities are distributed and realised across discourse units, thereby capturing the attentional state of discourse structure. More concretely, at any given point in a discourse, there is a *salient* entity that constitutes the focus of the discourse at that point. The notion of *salience* has been promoted by psychological studies of discourse (van Dijk and Kintsch, 1983; Givón, 1992); it describes the discourse elements that are more accessible in the memory of the reader/hearer, and therefore have a more prominent role in determining discourse coherence. In other words, as a reader/hearer processes a sentence/utterance, they build a mental representation of it in their memory in which some parts are more *active* than others and thus more anticipated to be encountered in the next sentences/utterances — these parts could be described as salient. Centering theory ranks the salience of entities according to their grammatical roles (GRs), where more prominent roles correspond to higher degrees of salience (e.g., *subject* > *object* > *indirect object* > *others*); this premise has been adopted by many entity-based theories (Brennan et al., 1987; Walker et al., 1994; Grosz et al., 1995; Kameyama, 1998). Other research has determined saliency based on other factors such as cognitive accessibility or familiarity (Prince, 1981; Gundel et al., 1993; Kameyama, 1998; Strube and Hahn, 1999), frequency (Barzilay and Lapata, 2008) or the surface positions of words (Gernsbacher and Hargreaves, 1988; Rambow, 1993).²

According to Centering theory, texts in which the same centers of attention are maintained in consecutive sentences are more coherent than those with repeated shifts from one entity to the other. I borrow two examples from Grosz et al. (1995):

- (a) (S1) John went to his favorite music store to buy a piano.
(S2) He had frequented the store for many years.
(S3) He was excited that he could finally buy a piano.
(S4) He arrived just as the store was closing for the day.

- (b) (S1) John went to his favorite music store to buy a piano.
(S2) It was a store John had frequented for many years.
(S3) He was excited that he could finally buy a piano.
(S4) It was closing just as John arrived.

¹Centering theory was initially proposed in 1986 by Grosz, B. J., Joshi, A. K., and Weinstein, S and widely circulated as a manuscript, then published in 1995.

²The surface positions of words is more useful in languages with free(r) word order such as German.

	$C_b(S_i) = C_b(S_{i-1})$ or undefined $C_b(S_{i-1})$	$C_b(S_i) \neq C_b(S_{i-1})$
$C_b(S_i) = C_p$	Continue	Smooth-Shift
$C_b(S_i) \neq C_p$	Retain	Rough-Shift

Table 2.1: Entity transitions in Centering theory.

Grosz et al. (1995) argue that example (a) is intuitively more coherent than (b) based on how the two entities (‘John’ and ‘store’) are introduced and realised. In (a), ‘John’ continues to be the focus of attention across all utterances while (b) keeps alternating focus between ‘John’ and ‘store’. More formally, each sentence S_i evokes a set of *forward-looking* centers (C_f s) and one *backward-looking* center (C_b). In S_i , the C_f s are ranked by salience, according to their grammatical roles (*subject* > *object* > *indirect object* > *others*), and the highest-ranked C_f is the *preferred* center (C_p). The highest-ranked element of S_{i-1} that is realised in S_i constitutes $C_b(S_i)$.³ There are 4 possible types of entity transitions across sentences; they are ranked from more coherent to less as: {**continue**, **retain**, **smooth-shift**, **rough-shift**} which I define in Table 2.1. For instance, in example (a) there is a **continue** transition between S_1 and S_2 as the center of attention is maintained; i.e., ‘John’ which is the subject and hence most salient entity in S_1 continues to have the same role in S_2 via pronominalisation. In contrast, the transition between S_1 and S_2 in example (b) is a **retain** one as the center of attention changes from ‘John’ in S_1 to ‘store’ in S_2 (via the pronoun ‘It’).

2.2 Traditional approaches to coherence modeling

In this section, I give an overview of coherence approaches that translate some of the aforementioned theories into computational models, using statistical NLP methods.

2.2.1 Semantic relatedness

Numerous coherence approaches were inspired by lexical cohesion that captures coherence in terms of repetition of words or using semantically related terms across text. I focus on two main approaches: *lexical chains* (Morris and Hirst, 1991) and *Latent Semantic Analysis (LSA)* (Landauer and Dumais, 1997).

Lexical chains A lexical chain is a sequence of semantically-related words that occurs in a span of text. For example, in the following paragraph from Morris and Hirst (1991), the underlined words form a lexical chain.

³This description of centers follows Brennan et al. (1987).

“In front of me lay a virgin crescent cut out of pine bush. A dozen houses were going up, in various stages of construction, surrounded by hummocks of dry earth and stands of precariously tall trees nude halfway up their trunks. They were the kind of trees you might see in the mountains.”

There are various ways to determine the candidate words for a chain including leveraging knowledge-bases such as thesauri (Morris and Hirst, 1991) or WordNet (Hirst et al., 1998), distributional co-occurrence of words (Marathe and Hirst, 2010), or topic-based models (Remus and Biemann, 2013). The strength of a chain could be estimated by a few properties such as length or degree of relatedness between words (Morris and Hirst, 1991; Barzilay and Elhadad, 1997; Hirst et al., 1998). A coherent text is expected to have strong chains. Accordingly, lexical chains were used to evaluate coherence in student essays (Somasundaran et al., 2014; Rahimi et al., 2015) and machine generated summaries (Lapata and Barzilay, 2005). They have also been leveraged in text summarisation since strong chains correspond to important parts of text that need to be extracted for the summary (Barzilay and Elhadad, 1997; Brunn et al., 2001; Silber and McCoy, 2002; Li et al., 2007; Ercan and Cicekli, 2008; Berker and Güngör, 2012; Lynn et al., 2018), as well as text segmentation tasks by capturing the linguistic structure of discourse and its topically related segments (Manabu and Takeo, 1994; Galley et al., 2003; Stokes et al., 2004; Marathe and Hirst, 2010; Tatar et al., 2013).

Latent Semantic Analysis (LSA) Spurring from the distributional hypothesis that words with similar meanings occur in similar contexts (Harris, 1954; Firth, 1957), LSA aims to capture the meaning of each word by constructing a matrix that represents its co-occurring words, then reducing its dimensionality to a vector via dimensionality reduction techniques (e.g., singular value decomposition (Berry et al., 1995)). A sentence vector is then generated by calculating the mean of the vectors of its words. Therefore, two sentences that contain semantically-related words are expected to have similar vectors. Foltz et al. (1998) adopted LSA to estimate local coherence by measuring the semantic similarity between adjacent sentences, where similarity is defined as the cosine similarity between their respective vectors. The overall document coherence is then computed by averaging its local similarity scores. LSA, and other semantic similarity methods that inherit from it, have been used to assess student writing quality (Wiemer-Hastings and Graesser, 2000; Landauer, 2003; Higgins et al., 2004; Higgins and Burstein, 2007; Yannakoudakis and Briscoe, 2012; Palma and Atkinson, 2018). The mental health domain has also leveraged LSA to assess discourses by patients who suffer from disordered speech, such as schizophrenia patients, and locate where the abrupt topic shifts occur (Elvevåg et al., 2007; Bedi et al., 2015; Iyer et al., 2018).

2.2.2 Probabilistic models

Probabilistic models derive from the underlying premise that we can predict the probability of a sentence, or generate it, based on its surrounding context. The overall coherence of a document is then approximated by combining the probabilities of all of its sentences. I categorise traditional probabilistic models into models that utilise lexical features or syntactic ones.

Lexicalised models Lapata (2003) estimated the probability of a sentence based on its previous sentence. They represented sentences as a set of features including verbs, nouns and grammatical dependencies, and calculated the probability of two sentences occurring consecutively using the cartesian product of their features. Barzilay and Lee (2004) approximated the probability of transitioning from one topic to another using a domain-specific content model, specifically a Hidden Markov Model (HMM), where each state corresponds to a distinct topic. In this model, a high transition probability corresponds to a more coherent text. Soricut and Marcu (2006) captured local coherence by adopting the IBM translation approach (Brown et al., 1993) that estimates the probability of a word appearing in a sentence conditioned on the words from its previous sentence. This is based on the idea that using certain words in a sentence triggers the usage of other words in the following sentences. Other probabilistic models rely on capturing coreference information. For instance, Elsner and Charniak (2008) created a *discourse-new* model that estimated the coherence of a document based on the type of mentions of its noun phrases (i.e., if it is a first mention or a subsequent one).

Syntax-based models The aforementioned probabilistic models are lexicalised, but unlexicalised generative models were also adopted. Louis and Nenkova (2012) proposed to capture the intentional structure of discourse (§2.1.3) using syntactic patterns, where the syntax of a sentence is represented by nodes at a specific level in its parse tree or a sequence of POS tags. They argued that sentences with similar syntactic structures are likely to have similar communicative goals that contribute to the purpose of the whole discourse. In other words, each sentence type (e.g., questions or definitions) has distinguishable syntax and therefore syntax could be used as a proxy for discourse intentions. In order to verify this hypothesis, they examined the grammatical production rules that co-occur in adjacent sentences in the WSJ and found that certain patterns often co-occur (further details are provided in §5.1). Louis and Nenkova (2012) estimated local coherence in terms of the probabilities of pairs of syntactic items occurring in adjacent sentences. They also implemented a global HMM-based coherence model where each state corresponds to similar syntactic constructions, thus presumably represents a discourse communicative goal, and transitions between states model the syntactic regularities of the discourse. Accordingly,

coherent texts in a specific domain that are expected to exhibit similar syntactic patterns should have similar transition probabilities that discriminate them from incoherent texts where those patterns are broken. The HMM-based approach surpassed the lexicalised content model of Barzilay and Lee (2004) and the EGrid model of Barzilay and Lapata (2008) that will be presented in the next section.

2.2.3 Entity-based methods

Entity-based approaches capture coherence by modeling the distribution and realisation of entities as a discourse unfolds. Some approaches directly translate Centering theory (§2.1.4) into a computational framework to evaluate local coherence. For example, Miltsakaki and Kukich (2000) manually annotated a corpus of student essays with entity transitions and found that rough shifts could be used as a proxy for essay incoherence. Furthermore, incorporating rough shifts in the *e-rater* essay scoring system (Burstein et al., 1998) significantly improved its performance. Rus and Niraula (2012b) on the other hand automatically detected the continue transitions of Centering theory and leveraged them to capture local coherence in student texts. Their method significantly correlated with human judgments for coherence.

Entity Grid (EGrid) Barzilay and Lapata (2005, 2008) built an unlexicalised model that captures local coherence by abstracting entity transitions in adjacent sentences; they referred to their model as Entity Grid (EGrid) representation. The grid is a matrix where rows represent sentences and columns refer to entities (which are the head nouns of NPs). Each cell $a_{i,j}$ denotes the grammatical role of the j^{th} entity in the i^{th} sentence and can take one of four values: *subject* ‘S’, *object* ‘O’, any *other* role ‘X’ or ‘-’ if it does not appear in the sentence. Fig. 2.1 shows an EGrid from Barzilay and Lapata (2008). Entity transitions are extracted from this grid according to a predefined length; for example, if the transition length is set to 2, there are 16 possible entity transitions (e.g., $\{S, S\}$, $\{O, X\}$, $\{X, -\}$,...etc.). Transition probabilities are then calculated from the EGrid to produce a feature vector for the text. For instance, in Fig. 2.1, the probability of the transition $\{S, -\} = 0.08$ (there are 75 transitions in the grid and $\{S, -\}$ occurs 6 times). A feature vector of length 16 will be generated by doing this calculation for each transition type, which Barzilay and Lapata (2008) used to train a Support Vector Machine (SVM; Vapnik, 1995) to discriminate between transitions in coherent documents vs. incoherent shuffled ones. Furthermore, Barzilay and Lapata (2008) investigated the impact of coreference (i.e., mapping coreferential entities to the same entity), salience (determined by entity frequency) and syntax (by creating another setup agnostic to syntax with only present (‘X’) or absent (‘-’) values for cells) and found that their impact is domain dependent. They applied their EGrid model on the tasks of ranking coherent documents against

<p>1 [The Justice Department]_s is conducting an [anti-trust trial]_o against [Microsoft Corp.]_x with [evidence]_x that [the company]_s is increasingly attempting to crush [competitors]_o.</p> <p>2 [Microsoft]_o is accused of trying to forcefully buy into [markets]_x where [its own products]_s are not competitive enough to unseat [established brands]_o.</p> <p>3 [The case]_s revolves around [evidence]_o of [Microsoft]_s aggressively pressuring [Netscape]_o into merging [browser software]_o.</p> <p>4 [Microsoft]_s claims [its tactics]_s are commonplace and good economically.</p> <p>5 [The government]_s may file [a civil suit]_o ruling that [conspiracy]_s to curb [competition]_o through [collusion]_x is [a violation of the Sherman Act]_o.</p> <p>6 [Microsoft]_s continues to show [increased earnings]_o despite [the trial]_x.</p>	<table border="1"> <thead> <tr> <th>Department</th> <th>Trial</th> <th>Microsoft</th> <th>Evidence</th> <th>Competitors</th> <th>Markets</th> <th>Products</th> <th>Brands</th> <th>Case</th> <th>Netscape</th> <th>Software</th> <th>Tactics</th> <th>Government</th> <th>Suit</th> <th>Earnings</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>S</td> <td>O</td> <td>S</td> <td>X</td> <td>O</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>2</td> <td>-</td> <td>-</td> <td>O</td> <td>-</td> <td>-</td> <td>X</td> <td>S</td> <td>O</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>3</td> <td>-</td> <td>-</td> <td>S</td> <td>O</td> <td>-</td> <td>-</td> <td>-</td> <td>S</td> <td>O</td> <td>O</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>4</td> <td>-</td> <td>-</td> <td>S</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>-</td> <td>S</td> <td>-</td> <td>-</td> <td>-</td> </tr> <tr> <td>5</td> <td>-</td> <td>S</td> <td>O</td> <td>-</td> </tr> <tr> <td>6</td> <td>-</td> <td>X</td> <td>S</td> <td>-</td> <td>O</td> </tr> </tbody> </table>	Department	Trial	Microsoft	Evidence	Competitors	Markets	Products	Brands	Case	Netscape	Software	Tactics	Government	Suit	Earnings	1	S	O	S	X	O	-	-	-	-	-	-	-	-	-	2	-	-	O	-	-	X	S	O	-	-	-	-	-	-	3	-	-	S	O	-	-	-	S	O	O	-	-	-	-	4	-	-	S	-	-	-	-	-	-	-	S	-	-	-	5	-	-	-	-	-	-	-	-	-	-	-	S	O	-	6	-	X	S	-	-	-	-	-	-	-	-	-	-	O
Department	Trial	Microsoft	Evidence	Competitors	Markets	Products	Brands	Case	Netscape	Software	Tactics	Government	Suit	Earnings																																																																																												
1	S	O	S	X	O	-	-	-	-	-	-	-	-	-																																																																																												
2	-	-	O	-	-	X	S	O	-	-	-	-	-	-																																																																																												
3	-	-	S	O	-	-	-	S	O	O	-	-	-	-																																																																																												
4	-	-	S	-	-	-	-	-	-	-	S	-	-	-																																																																																												
5	-	-	-	-	-	-	-	-	-	-	-	S	O	-																																																																																												
6	-	X	S	-	-	-	-	-	-	-	-	-	-	O																																																																																												

Figure 2.1: EGrid example from Barzilay and Lapata (2008). The left hand side displays a text with grammatical annotations for entities and the right hand side depicts the grid representation for the text where cells correspond to grammatical roles (subjects (S), objects (O), or neither (X)), or (-) if the entity does not appear in the sentence.

their incoherent permuted versions, evaluating the coherence of summaries and readability assessment.

The EGrid has become a de facto coherence model that was further extended and widely adopted in discourse-related applications. Elsner and Charniak (2008) were able to significantly improve their discourse-new model (§2.2.2) by incorporating EGrid transition probabilities. Elsner and Charniak (2011b) extended EGrids with entity-specific features such as: whether an entity has a proper mention (i.e., is realised by a proper noun), whether it has a singular mention, its Named Entity label and number of modifiers, in addition to other coreference domain-specific features. They also expanded the definition of entities to include non-head mentions in NPs. Their enhanced EGrid version outperformed the basic one in coherence discrimination and sentence insertion tasks. Guinaudeau and Strube (2013) adapted EGrids in a graph-based framework where there are two sets of nodes: sentences and entities. An entity is linked to the sentence it appears in via an edge weighted by its grammatical role, and entity transitions between sentences are modeled by a projection graph, where two sentences are connected if they share the same entity. EGrids were also combined with HMM generative approaches in order to capture both local and global coherence via log-linear models (Soricut and Marcu, 2006) or learning local and global features jointly (Elsner et al., 2007). Other models extended the EGrids with semantic relatedness features between entities that do not necessarily entail coreference, where semantic relations are extracted from knowledge bases (Filippova and Strube, 2007; Zhang et al., 2015). EGrids were also adapted to other languages; Cheung and Penn (2010a) substituted grammatical roles with *topological fields* that capture high-level clausal structure in German and showed that this approach further boosts performance on detecting permuted texts.

2.2.4 Coherence relation models

Discourse research has also exploited coherence/rhetorical relations (§2.1.2) in computational models for coherence assessment. Lin et al. (2011) leveraged the PDTB coherence relations to build a matrix representation of text with rows corresponding to sentences and columns to terms that take role as an argument in the predicate-argument style relations. In the matrix, a cell $a_{i,j}$ corresponds to the coherence relation type and argument in which term i takes part in sentence j ; e.g., if a term is part of Arg1 in a comparison relation the cell will be annotated with ‘Comp.Arg1’. Similar to Barzilay and Lapata (2008), coherence is calculated based on the probability of the relation/argument transitions in the matrix. They were able to gain further improvements in coherence ranking by combining their model features with EGrid features. PDTB-style discourse relations were also utilised in graph-based models (Guinaudeau and Strube, 2013) and combined with entity graphs for readability assessment (Mesgar and Strube, 2015). Feng et al. (2014) adopted an approach similar to Lin et al. (2011) but using RST relations and proved their efficacy over PDTB relations and EGrids in coherence ranking and detecting organisation in student essays. RST-based models have been utilised in various domains such as coherence ranking in Brazilian Portuguese texts (Dias et al., 2014) and assessing coherence in student writing (Burstein et al., 2013; Huang et al., 2018).

2.3 Deep learning representations

In the previous section, I gave an overview of traditional approaches to coherence modeling that spurred from different discourse theories and frameworks. Before moving to describing neural coherence approaches, I present, in this section, methods leveraged by neural coherence models (and neural networks in general) to encode textual units such as words or sentences.

2.3.1 Neural encoders

A large number of popular architectures have been used in NLP deep learning models to encode representations of linguistic units. I here discuss three main approaches that are referred to throughout this thesis.

Recurrent Neural Networks (RNN) RNNs (Elman, 1990) are designed to model sequential information and therefore are well-suited for NLP. An RNN processes a sequence (of characters, words, sentences, etc.) element by element and applies the same functions to each element at each time step (t). The model calculates a hidden representation (h_t) at each time step using the input element at this step (x_t) and the hidden representation

of the previous element (h_{t-1}):

$$h_t = \tanh(U \cdot x_t + W \cdot h_{t-1} + b) \quad (2.1)$$

where $U \in \mathbb{R}^{k \times d}$ and $W \in \mathbb{R}^{d \times d}$ are weight matrices and $b \in \mathbb{R}^d$ is a bias vector; k is the length of the input vector x_t and d is a hyperparameter indicating the size of the hidden layer. Traditionally, the initial hidden representation used at the first time step ($h_{t=0}$) is initialised with zeros. Despite their ability to capture sequential information, in practice, RNNs struggle with modeling long-term dependencies. Therefore, enhanced models have been introduced to solve this problem such as Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (GRU) (Cho et al., 2014). I focus on LSTM in my work.

LSTM extends the vanilla RNN model by adding a mechanism to control what the network should remember or forget, in the long term, at each time step. This is achieved by calculating a cell state (c_t) that represents the network memory at t , a ‘forget gate layer’ (f_t) to control what to keep from the previous cell state (c_{t-1}), an ‘input gate layer’ (i_t) to decide what values to update in c_t and an ‘output gate layer’ (o_t) to decide what parts of c_t to output:

$$\begin{aligned} i_t &= \sigma(x_t \cdot U_i + h_{t-1} \cdot W_i + b_i) \\ f_t &= \sigma(x_t \cdot U_f + h_{t-1} \cdot W_f + b_f) \\ o_t &= \sigma(x_t \cdot U_o + h_{t-1} \cdot W_o + b_o) \\ c_t &= c_{t-1} \odot f_t + \tanh(x_t \cdot U_c + h_{t-1} \cdot W_c + b_c) \odot i_t \\ h_t &= \tanh(c_t) \odot o_t \end{aligned} \quad (2.2)$$

where $U \in \mathbb{R}^{k \times d}$, $W \in \mathbb{R}^{d \times d}$ and $b \in \mathbb{R}^d$ are the network’s learned parameters; σ is the sigmoid function, \odot is the Hadamard product and $c_{t=0}$ and $h_{t=0}$ are initialized with zero vectors. Equations 2.2 could be abstracted as:

$$h_t = LSTM(x_t, h_{t-1}) \quad (2.3)$$

Equations 2.1 and 2.2 encode the hidden vectors based on their previous context, ignoring what comes next. This problem is addressed by applying *Bidirectional RNNs* that, in addition to the previous context, build a hidden representation at t based on the hidden representation at the next time step ($t+1$). This will result in two vectors for t encoding its left and right contexts, which could then be concatenated or aggregated by other functions such as averaging or multiplication. For instance in a Bidirectional LSTM

(Bi-LSTM), h_t could be calculated by:

$$\begin{aligned}\vec{h}_t &= LSTM(x_t, \vec{h}_{t-1}) \\ \overleftarrow{h}_t &= LSTM(x_t, \overleftarrow{h}_{t+1}) \\ h_t &= [\vec{h}_t, \overleftarrow{h}_t]\end{aligned}\tag{2.4}$$

Convolutional Neural Network (CNN) CNN (LeCun et al., 1998) is a neural architecture that extracts local features from input; it was initially proposed for image processing then successfully imported to NLP (Collobert and Weston, 2008; Kim, 2014). The key function in a CNN is *convolution* that slides a filter of weights over windows of local context to extract local features. More concretely, a filter $W \in \mathbb{R}^{k \times l}$ slides over the input text $x \in \mathbb{R}^{k \times n}$ and at each location, an element-wise multiplication between W and a window of size l in x is applied. The resulting matrix is then summed up to indicate feature h_i for this window, where k is the vector dimensionality of an input feature (e.g., character, word,...etc), l is the filter length and n is the input length. The convolution operation could be followed by a non-linearity:

$$h_i = \tanh([x_i; \dots; x_{i+l-1}] * W)\tag{2.5}$$

Here, $i \in \{1, \dots, n - l + 1\}$ and $*$ is the linear convolutional operation. The local features extracted at different positions (h_i) form a *feature map*. Multiple filters could be applied to extract various feature maps. In order to highlight the important features and extract global features from the local ones, a max pooling operation is applied to each feature map to select the highest-value feature(s). Other pooling operations could also be applied such as average or L2 norm (Goodfellow et al., 2016, p. 335). Unlike RNNs, pooling allows CNNs to become transitional invariant, which means that they are not sensitive to the order of input features except locally (in the window where the convolutional filter is applied).

Transformer Recently, a transformer model (Vaswani et al., 2017) was proposed to learn the relations between input features using a self-attention mechanism. A transformer encoder consists of *Multi-Head Attention* followed by a feed forward layer, where each attention head aims to measure the importance of each word in relation to the words in the input sequence. A transformer decoder has the same architecture as the encoder with an extra encoder-decoder attention layer. An attention head first maps each input feature x_t (e.g., word) to three vectors: *query* $\in \mathbb{R}^{d_k}$, *key* $\in \mathbb{R}^{d_k}$ and *value* $\in \mathbb{R}^{d_v}$ by multiplying x_t with learned weight matrices W_q , W_k and W_v . The vectors for the whole input sequence x could be compacted in matrices Q for query, K for key and V for value. Q and K are then used to calculate a score between each two input features (including a feature and

itself) and attention is calculated by multiplying this score to the value vector:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.6)$$

An input feature will, therefore, have a number of output vectors representing its relation with each input feature and these vectors are then added to form one vector per feature. With multiple attention heads that use different sets of query, key and value vectors, the encoder calculates multiple representations per input feature which are concatenated and multiplied with a learned weight, then fed to a feed forward layer to produce the final output. In contrast to recurrent networks, a transformer computes the output vectors at different input positions in parallel and the execution of the multiple heads is also achieved in parallel, resulting in a more time-efficient model.

2.3.2 Word representations

Most NLP tasks fundamentally process words, and therefore researchers have devoted much attention to investigating how to represent them. Initialising word representations/embeddings from pre-trained spaces that capture aspects of their syntax and semantics has become the standard initialisation method for input words in neural networks. The idea is to pre-train word embeddings on an unsupervised task using large unlabelled corpora to capture their distributional properties, then use them to initialise the network. This allows bootstrapping networks from a semantically-rich space, which is particularly useful in low-resource tasks, instead of initialising word vectors randomly or as one-hot vectors. We next discuss the two main types of embeddings in the literature: standard and contextualised.

Standard word embeddings. In standard embeddings, each word in the pre-trained space is represented by a single context-independent vector. For instance, the word *play* in “I *play* football” and in “Yesterday I watched a *play*” is mapped to the same vector, regardless of the different syntactic categories it belongs to. Throughout this thesis, I refer to the context-independent word embeddings as *standard* word embeddings. These embeddings are used to initialise neural networks for downstream tasks, and could either be kept constant during training, or fine-tuned to be more task-specific.

There is a wide-range of approaches to building distributional word embeddings; LSA is one of the early approaches that relies on word co-occurrence and dimensionality reduction techniques. Afterwards, neural models have been widely adopted to learn such representations (Bengio et al., 2003; Mikolov et al., 2011; Huang et al., 2012). Mikolov et al. (2013a) proposed `word2vec` models that either employ *Continuous Bag-of-Words* approach to predict a center/target word based on its context, or *Continuous Skip-gram*

approach to predict the surrounding context given the target word. Mikolov et al. (2013c) released pre-trained embeddings that have become ubiquitous in NLP tasks and were trained as a Skip-gram model on a corpus of Google News articles that contain around 100B words.⁴ Another widely-used set of pre-trained embeddings are the Global Vectors (GloVe) (Pennington et al., 2014) trained on Wikipedia articles;⁵ GloVe models word co-occurrences, similar to LSA, but using a matrix factorisation method that leverages the log probability of co-occurrences. Other embeddings rely on word position information (Mnih and Kavukcuoglu, 2013), subword properties (Bojanowski et al., 2017) or idiomatic phrases (Mikolov et al., 2013c), or combine all these approaches in one model, such as **fastText** embeddings⁶ (Mikolov et al., 2018) that are pre-trained on the Common Crawl corpus of 600B tokens.⁷ Cross-lingual word embeddings have also been proposed to transfer knowledge across different languages (Zou et al., 2013; Gouws et al., 2015; Luong et al., 2015).

Contextualised word embeddings. Instead of representing each word as a fixed vector, contextualised word embeddings represent words as a function of their context; this way, the same word will be mapped to different representations according to its surrounding context (i.e., the sentence it appears in). This dynamic representation helps disambiguate word senses (e.g., ‘play’ in the aforementioned example), and thus builds a richer semantic space. There are various contextualised models that emerged since 2017, starting with Context Vectors (CoVe) (McCann et al., 2017) learned from a machine translation encoder that is attentional and LSTM-based, followed by many more (Howard and Ruder, 2018; Akbik et al., 2018; Radford et al., 2018, 2019; Yang et al., 2019b). In this thesis, I focus on *Embeddings from Language Models (ELMo)* (Peters et al., 2018) and *Bidirectional Encoder Representations from Transformers (BERT)* (Devlin et al., 2019) as two of the most successful models that have boosted performance on a variety of NLP tasks, such as question answering (Rajpurkar et al., 2018; Yang et al., 2019a), summarisation (Gehrmann et al., 2018; Liu and Lapata, 2019) and machine translation (Zhu et al., 2020). There are two approaches to leveraging these models in downstream tasks: *feature-based*, where static features are extracted from the pre-trained models and fed to a new task-specific model, or *fine-tuning*, where the pre-trained model is re-trained and fine-tuned to perform the target task. I use the first in this thesis.

ELMo utilises a deep LSTM-based forward and backward language model to create three layers of representation for each input word (an input word embedding layer and two Bi-LSTMs). More specifically, it stacks two forward LSTMs and another two backward

⁴<https://code.google.com/archive/p/word2vec/>

⁵<https://nlp.stanford.edu/projects/glove/>

⁶<https://fasttext.cc/docs/en/english-vectors.html>

⁷<http://commoncrawl.org/2017/06/>

LSTMs and concatenates the hidden representations for each layer from both directions at each time step. The bottom word embedding layer leverages character-based representations, enabling the model to handle out-of-vocabulary words. ELMo is pre-trained on the 1B Word Benchmark corpus (Chelba et al., 2014). After pre-training, the model layers could be used (individually or combined) to initialise neural networks designed to perform various tasks. Peters et al. (2018) suggested calculating task specific weighting of the three layers to form a single vector ($ELMo_t$) for each word:

$$ELMo_t = \gamma \sum_{k=1}^L v_k h_{tk}^{LM} \quad (2.7)$$

where L is the number of hidden layers (3 in that case), v_k is a weight assigned to the k -th layer, h_{tk}^{LM} is the representation of the t -th word at the k -th layer, and γ is a weight to scale the whole ELMo vector based on the task. They also motivated only taking the last layer, following Peters et al. (2017), or simply averaging the three layers, which means fixing the value of γ at 1 and assigning equal weights v_k to all the layers ($v_k = 1/3$). Peters et al. (2018) empirically investigated, via intrinsic evaluation, the linguistic properties encapsulated by each layer of representation and found that syntactic properties are better captured by lower layers whereas higher layers better represent semantic features.

On the other hand, **BERT** pre-trains a language model using multi-head transformer encoders. What is special about BERT is that it builds a language model by performing bidirectional training for the transformer model, meaning that each word representation relies on the context on its left and right. This is in contrast to ELMo that learns the right and left contexts independently then concatenates their resulting representations. In order to allow bidirectional language modeling, BERT randomly masks some percentage of the input tokens and learns to predict these masked tokens. BERT utilises WordPiece embeddings (Wu et al., 2016), and accordingly each word is tokenised into subwords (e.g., ‘embeddings’ is tokenised into [‘em’, ‘##bed’, ‘##ding’, ‘##s’]). The model is pre-trained on the BooksCorpus (800M words) (Zhu et al., 2015) and English Wikipedia (2,500M words). It is also pre-trained on a sentence prediction task where the model must predict whether two sentences are consecutive. Devlin et al. (2019) created two pre-trained BERT models: (1) BERT_{BASE} with 12 layers (i.e., transformer encoders), 12 attention heads, feedforward layers of dimension 768 each and (2) BERT_{LARGE} with 24 layers/encoders, 16 attention heads, feedforward layers of dimension 1024 each. Accordingly, each input word is mapped to either 12 or 24 vectors that could be used individually or combined, similar to ELMo. For each BERT model, there is a cased (i.e., lowercased) and uncased version. A large body of work has investigated the linguistic information encoded in BERT layers. Some studies have shown that syntactic information is better captured in the middle layers (Hewitt and Manning, 2019; Jawahar et al., 2019), semantic features are spread

across all the layers (Tenney et al., 2019a) and the middle layers are the most transferable (i.e., perform better on different tasks) (Liu et al., 2019b).

2.3.3 Sentence representations

The meaning of a sentence is interpreted in terms of the meaning of its words, thus it is important to learn composition functions over word representations (standard or contextualised) to capture sentence meaning. Linear transformations to word vectors (e.g., addition, multiplication or average) have been investigated to build phrase and sentence representations (Mitchell and Lapata, 2008; Blacoe and Lapata, 2012). Other more complex approaches have utilised neural encoders, such as the ones presented in §2.3.1, to encode the sequence of words in a sentence. The output of these encoders are latent representations (i.e., a hidden representation for each word in RNNs and transformers or for each local window of words in CNNs) that are aggregated into a fixed-length vector representing the sentence meaning. In RNNs, the hidden representation of the last word has commonly been used to represent the whole sentence (Cho et al., 2014; Sutskever et al., 2014). Similarly in BERT, a transformer model, a sentence can be represented by the output vector corresponding to the special token ‘[CLS]’ used to mark the beginning of a sequence (Devlin et al., 2019; Liu et al., 2019c). Alternatively, as mentioned before, different operations, such as addition, multiplication or average, could be applied to the hidden word representations to collapse them into a single sentence embedding, instead of taking one representation (e.g., the hidden state of the last word or the ‘[CLS]’ vector) to encapsulate the whole sentence. However, these operations give equal weight to all words despite the fact that some words might contribute more to the sentence meaning. Therefore, *attention* mechanisms were introduced to mitigate this problem and help the network focus on the important words. In NLP, attention was initially proposed for sequence-to-sequence (seq2seq) machine translation models to allow the decoder to learn what parts of the input sentence to attend to while generating the output sentence (Bahdanau et al., 2015). Attention has also been used as an aggregation mechanism to combine, for instance, word representations into one single sentence vector, while highlighting the words that are more important to the final network prediction (Yang et al., 2016b; Lin et al., 2017). In order to generate a sentence vector (s) by attending to its word hidden representations, the following equations are applied (Bahdanau et al., 2015):

$$\begin{aligned}
 u_t &= \tanh(Wh_t) \\
 a_t &= \frac{\exp(vu_t)}{\sum_t \exp(vu_t)} \\
 s &= \sum_t a_t h_t
 \end{aligned}
 \tag{2.8}$$

where h_t is the hidden representation of the word at position t and W and v are learnable parameters. Finally, a sentence could be encoded independently (i.e., as a function of its own words) or by also relying on its surrounding sentences (Kiros et al., 2015; Li and Jurafsky, 2017).

Document/Paragraph representations In order to build higher representations, e.g., for paragraphs and documents, the same sentence encoding methods could be used. A document representation could be generated in a hierarchical way, by applying neural encoders and aggregation methods to sentence embeddings that were constructed earlier from word vectors (Li et al., 2015; Yang et al., 2016b). Alternatively, a document vector could access word-level representations directly, ignoring sentence boundaries, e.g., by averaging the hidden representations resulting from applying an LSTM over the whole sequence of words in the document (Taghipour and Ng, 2016), or generating a document vector by extending a `word2vec` model (`doc2vec`; Le and Mikolov, 2014). In this thesis, I leverage a hierarchical approach and build a document representation from its sentence vectors.

2.4 Neural approaches to coherence modeling

Until 2014, coherence modeling relied on traditional NLP approaches (§2.2). Since then however, due to the rapid advances in deep learning, neural models have been widely adopted in coherence modeling, outperforming previous approaches. In this section, I discuss different neural coherence systems to put my work in context. I categorise the systems into discriminative and generative.

2.4.1 Discriminative approaches

Supervised discriminative coherence models are trained to discriminate between labelled coherent and incoherent instances. My MTL approach (Chapter 3) falls under this category of models. I divide them into end-to-end or entity-based.

End-to-end The earliest neural coherence model is the *Local Coherence (LC)* model (Li and Hovy, 2014) that leverages a window approach, i.e., a CNN. The model first uses an RNN or a recursive neural network (Socher et al., 2011) to construct sentence vectors, that are concatenated to build a clique embedding.⁸ A filter of weights then slides over clique representations to score them and the resulting scalar score is mapped to $[0, 1]$ using a sigmoid function. The final coherence score of text is calculated by multiplying

⁸A clique is a sequence of neighbouring sentences.

its composing clique scores. Li and Jurafsky (2017) adopted a similar approach but used an LSTM to construct sentence embeddings and obtained the overall coherence score by averaging clique scores.

Cui et al. (2017) utilised a CNN to build sentence representations (i.e., applying convolution over word embeddings followed by max pooling). A clique representation (of three sentences) is learned in turn by first calculating two similarity scores (with a bilinear function) between the the first and second sentences, and the second and third sentences and then concatenating these scores with a concatenation of the three sentences. Scoring the cliques and approximating a document coherence score follows Li and Jurafsky (2017).

Calculating similarity scores between adjacent sentences to capture local coherence has also been leveraged by Mesgar and Strube (2018). They applied an LSTM to generate hidden states for words in each sentence, then selected the states that were most similar (using the dot product) for every adjacent two sentences. These two hidden states were then averaged into feature vector f_i , then the similarity between each consecutive feature pair (f_i and f_{i+1}) was calculated to measure the degree of continuity in the input document. A coherent document is expected to have high similarity scores.

Moon et al. (2019) integrated a local and global model for coherence. Their local model generates hidden representations (h_i) for sentences with a Bi-LSTM with an explicit language model loss. Subsequently, a bilinear operation is applied to project each two adjacent sentences into a vector (v_i) which represents local context. In order to capture global coherence, a *light weight* CNN (Wu et al., 2019) is applied over the vectors h_i , which is a special type of CNN with reduced network parameters. The CNN is then followed by average pooling to produce a global document vector u . Finally, in order to combine the local and global models, u is concatenated with each pair of local vectors v_i and v_{i+1} and the output is fed to a linear layer to predict a local score; the overall score for the document is the sum of its local scores. Moon et al. (2019) showed the efficacy of their model on coherence ranking in addition to evaluating local coherence by training and testing the model to detect documents where one or more window of sentences are permuted while the rest of the document is kept intact.

Xu et al. (2019) developed a Local Coherence Discriminator (LCD) that leverages a generative approach to build sentence representations, then adds a discriminative layer to distinguish between coherent and incoherent pairs of sentences. The underlying premise of the LCD model is that a coherent pair of adjacent sentences (s_i, s_{i+1}) should be ranked higher than an incoherent one (s_i, s'). Incoherent pairs are created by negative sampling, where in each training epoch, for each document, 50 triplets (s_i, s_{i+1}, s') are sampled, where (s_i, s_{i+1}) constitutes a coherent pair taken from the document and (s_i, s') is an incoherent pair where s' is randomly selected from the same document. This sampling strategy allows the model to learn from a large space of negative examples without expensive computations.

As for building the input sentence representations, Xu et al. (2019) employed three encoders: (1) an RNN language model (LCD-L), (2) *InferSent*: a sentence encoder by Conneau et al. (2017), trained in a supervised way on the large Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015), (3) averaging the GloVe vectors of the words in the input sentence. After generating sentence representations, linear transformations are applied to the two sentence vectors in positive or negative pairs; the transformations are: concatenation, element-wise difference, element-wise product and absolute value of element-wise difference. The outputs of these transformations are then concatenated to produce vector O that is fed to a one-layer MLP to predict a local coherence score for each sentence pair:

$$O = [S, T, S - T, S * T, |S - T|] \quad (2.9)$$

where S corresponds to the first sentence representation (i.e., s_i) and T to the second (i.e., s_{i+1} or s'). The same model is trained in the reverse direction (i.e., given the input pair in the reverse order: (s_{i+1}, s_i) for positive pairs and (s', s_i) for negative ones). The local coherence score of an input pair is the average of its two scores generated by the forward and backward models and the final overall score of a document is calculated by averaging its local scores. The network is trained in a pairwise fashion and optimises a *margin loss* that aims to maximise the scores of the positive pairs as well as minimise the scores of the negative ones. Xu et al. (2019) applied their model to the WSJ corpus in addition to open-domain experiments of Wikipedia articles; their best overall results were using an RNN encoder (LCD-L). Concretely, LCD-L applies an RNN over the word vectors in each input sentence and optimises the difference between the conditional log likelihood of a sentence given its previous context and the language model probability for generating the sentence. The final sentence vector is generated by maxpooling the hidden states of its words. The LCD-L model is the current published state-of-the-art on the WSJ in ranking an original document higher than **its** permuted versions. We widely utilise the LCD approach in this thesis and enhance it with contextualised embeddings as will be discussed in Chapter 4.

Entity-based The efficiency of EGrid models in capturing entity distributions throughout text inspired researchers to adapt them for neural models. Tien Nguyen and Joty (2017) argue that utilising entity-based features in neural frameworks can enhance performance by capturing “long range entity transitions”. More specifically, traditional non-neural entity-based approaches define a length (l) for entity transitions of G different GRs, meaning that G^l probabilities must be calculated which exponentially grows with longer transitions. Tien Nguyen and Joty (2017) developed a CNN-EGrid model that applies a CNN over EGrid transitions of input texts. The CNN slides multiple filters of weights to extract feature maps that represent high-level entity-transition features, followed by a max

pooling function to focus on the important features. Since the filters can have large sizes (they use a size of 5 – 8), long-range transitions can be captured more efficiently than previous entity-based methods. Training is performed in a pairwise fashion where the model takes a pair of documents as input, specifically a coherent document and its incoherent counterpart, and optimises the margin loss similar to Xu et al. (2019). Furthermore, they extended the model by attaching three entity-specific features (Elsner and Charniak, 2011b) to the distributed representations of entities: named entity type, salience (represented as the occurrence frequency of entities) and a binary feature indicating whether the entity has a proper mention. I refer to this extended model as *CNN-EGrid_{ext}*.

CNN-EGrid is agnostic to the lexical properties of entities which result in a model that is unable to differentiate between transitions of various entities. Extending the model with entity-specific features (CNN-EGrid_{ext}) can mitigate this problem, yet requires an additional feature extraction step which is less generalisable to low-resource languages (Joty et al., 2018). To resolve this, Joty et al. (2018) further extended the CNN-EGrid model with lexical information about the entities. More concretely, they represented each entity as a combination of its word embedding, retrieved from a pre-trained space, and its GR (S, O, X). For instance, if “Obama” appears as a subject in one location and an object in another, there will be two different representations for it in the input embedding matrix: Obama-S and Obama-O. Joty et al. (2018) managed to outperform CNN-EGrid_{ext} in coherence ranking on the WSJ dataset without including the three entity-specific features in their model. I refer to this lexicalised version as *CNN-EGrid_{lex}*.

2.4.2 Generative approaches

Unsupervised learning has also gained much attention in coherence modeling and a plethora of generative approaches have been developed. Although I do not build generative models in this thesis, in this subsection I discuss this type of models to put my work in a wider context. Generative models learn to produce a coherent sequence of sentences by generating one sentence at a time, based on the previously generated sentences. Li and Jurafsky (2017) proposed a generative LSTM-based seq2seq model, similar to machine translation models (Sutskever et al., 2014), that learns to predict a sentence based on its previous and next sentences. The model is only trained on original documents to maximise the likelihood of coherent contexts. In order to enhance the model with document global information, the sentence decoder utilises topic vectors learned either using Latent Dirichlet Allocation (LDA) (Blei et al., 2003) (which extracts latent topic vectors (i.e., topic distributions) from training data) or automatically from the training data using a hierarchical LSTM model (i.e., an LSTM over words to generate sentence vectors and another LSTM over these vectors to generate a document vector). Li and Jurafsky (2017) showed that discriminative approaches (e.g., the LC model) perform better in in-domain coherence ranking, but their

generative model generalises better to open-domain experiments.

Subsequent unsupervised models have been trained to predict sentence order and have been particularly popular in information ordering, where the model is asked to retrieve the original order of a permuted set of sentences. More formally, given a set of n sentences $s = [s_1, s_2, \dots, s_n]$ with the order $o = [o_1, o_2, \dots, o_n]$, the model should learn the correct order $o^* = [o_1^*, o_2^*, \dots, o_n^*]$. *Pointer Networks (Ptr-Net)* (Vinyals et al., 2015) have been widely used in this task. Ptr-Nets are well-suited for sorting variable sized sequences because they generate a discrete output corresponding to positions in the input sequence by using attention as a pointer to select input positions. A Ptr-Net based decoder utilised in encoder-decoder coherence models predicts an order (\hat{o}) for input sentences by producing a probability distribution over the sentences at each time step:

$$P(o|s) = \prod_i^n P(o_i | o_{i-1}, \dots, o_1, s) \quad (2.10)$$

The predicted order with the highest coherence probability is then selected as the final order:

$$\hat{o} = \operatorname{argmax}_o P(o|s) \quad (2.11)$$

At training time, the decoder is given the sentences (s) in the correct order, while at test time, at each time step the input is the predicted output of the previous time step.

Several approaches have employed this Ptr-Net-based ordering strategy (Gong et al., 2016; Logeswaran et al., 2018; Cui et al., 2018; Wang and Wan, 2019; Oh et al., 2019; Yin et al., 2019b, 2020), with the core difference being the architecture of the encoder and decoder. Gong et al. (2016) tested three encoding strategies: a Bag-of-Words, an LSTM and a CNN, and got the best overall results with the LSTM approach in the task of sentence ordering. Logeswaran et al. (2018) also employed an LSTM-based encoder-decoder framework for sentence ordering; they specifically adopted the architecture of Vinyals et al. (2016). Their model first creates sentence embeddings using an LSTM, then processes these embeddings with another LSTM encoder, where at each time step, attention weights are calculated based on the sentence embeddings and the current hidden state. These weights are used as input for the next time step. The decoder is a Ptr-Net that is similar to the encoder, but uses sentence embeddings as input instead of using attention weights.

Oh et al. (2019) used an LSTM encoder and a transformer decoder. They also used LDA to learn latent topic vectors for sentences and paragraphs to capture local and global context respectively. Each LSTM sentence representation is aggregated with its topic vector and its paragraph topic vector, using a linear transformation followed by a non-linear activation, to generate *topic-sensitive* sentence vectors. Finally, the decoder is a transformer-based network that leverages a Ptr-Net to predict sentence order. In contrast, Cui et al. (2018) used an LSTM and a transformer network as the encoder and

an LSTM pointer-based decoder. Specifically, for the encoder, they first used the LSTM to encode sentences then applied a transformer on top to encode the paragraph, where the paragraph representation is the average of the output of the transformer last self-attention layer. The LSTM decoder is then initialized with this encoded paragraph vector. Yin et al. (2019b) extended the work of Cui et al. (2018) by replacing their encoder with a sentence-entity graph (Guinaudeau and Strube, 2013) encoded with a recurrent neural graph (Zhang et al., 2018).

Instead of using a pointer network decoder, some approaches predicted a position for each sentence independently, without referring to the input sequence. Kumar et al. (2020a) predicted real-valued positions with a feed forward network and used a pre-trained BERT encoder. Specifically, each sentence is scored with the feed forward layer and the scores are sorted to output the final order. For example, for a document of 5 sentences the predicted scores $[y_1, y_2, y_3, y_4]$ could be $[0, 0.25, 0.50, 0.75, 1.0]$. This approach outperformed previous Ptr-Net based approaches in the information ordering task. Similarly, Bohn et al. (2019) used a position model yet predicted a discrete distribution over possible positions for each sentence. More concretely, they first applied a stacked Bi-LSTM where the input at each time step is the concatenation of the pre-trained input word embedding, the average of pre-trained word embeddings for the whole document (to capture global context) and the difference between both. The LSTM output is then fed to a softmax layer to predict a probability distribution for each sentence over a pre-defined number of quantiles; e.g., if the number of quantiles is 4, the model predicts which quarter of the document the sentence belongs to.

2.5 Multi-task learning

In this section, I give a brief overview of MTL which forms the basis of my main model in this thesis. Machine learning models typically focus on performing a *single* task and accordingly tune their parameters to optimise a particular loss function (e.g., mean square error or binary cross entropy). This *Single-Task Learning (STL)* setup ignores useful information that could be learned from other related tasks. In contrast, *Multi-Task Learning (MTL)* is “an approach to inductive transfer that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias” (Caruana, 1997). For instance, tasks such as POS tagging, Semantic Role Labelling (SRL) and Named-Entity Recognition (NER) could be performed together as POS tags are often used as features for SRL and NER (Collobert and Weston, 2008). MTL is a type of transfer learning where two or more tasks are learned simultaneously; the tasks are either of the same importance, or there is a main task that the model focuses on and one or more auxiliary tasks to help improve performance on the main one. The

model optimises a cost function that combines the loss functions of the different tasks; one simple approach is to calculate a weighted sum over the individual functions:

$$Loss = \sum_{i=1}^T \lambda_i L_i \quad (2.12)$$

where T is the number of tasks, λ_i is the weight of the i -th task and L_i is its loss function. All tasks are either assigned equal weights (e.g., $\lambda_i = 1$) or different ones to give more attention to particular tasks. For example, if we have two tasks – a main one that we want to focus on with loss L_1 and an auxiliary task with loss L_2 – then typically $\lambda_1 > \lambda_2$. The weights (λ_i) are tuned as hyperparameters of the network to find the best balance between tasks.

MTL is achieved by parameter sharing between tasks which can be categorised as *hard sharing* (Caruana, 1993) or *soft sharing* (Duong et al., 2015; Yang and Hospedales, 2017). In hard parameter sharing, which I use in this work, model hidden layers are shared between tasks while keeping the output layers task-specific. On the other hand, in soft parameter sharing, each task has its own network and the distance between the parameters of the networks is regularised to be similar (Ruder, 2017).

In MTL, each task is associated with a dataset that can be shared between, or different from, other tasks. In the latter case, the model alternates between the tasks, during training, using various methods, including randomly selecting a task and randomly sampling a batch from its corresponding dataset at each training epoch (Søgaard and Goldberg, 2016), or iterating over all the tasks and processing their full training datasets in each epoch (Hashimoto et al., 2017).

A large body of work has focused on applying MTL to **word-level** tasks such as POS tagging, named entity recognition, syntactic chunking, coreference resolution and grammatical error detection (Collobert and Weston, 2008; Collobert et al., 2011; Plank et al., 2016; Søgaard and Goldberg, 2016; Yang et al., 2016a; Rei, 2017; Rei and Yannakoudakis, 2017; Sanh et al., 2019). Some approaches supervise all the tasks at the same level (typically the outermost layer) (Collobert et al., 2011; Rei, 2017), while others investigate which level in the network hierarchy is best for supervision. For instance, Søgaard and Goldberg (2016) argued that POS tags are better predicted at lower layers in a multi-layer Bi-LSTM network, and Sanh et al. (2019) leveraged hierarchical inductive bias between tasks in a multi-layer Bi-LSTM model by supervising named entity recognition at the first level, entity mention detection at the second and coreference resolution and relation extraction at the third.

MTL has also been utilised to perform **sentence-level** tasks such as subjectivity evaluation and sentiment analysis (Liu et al., 2016; Yu and Jiang, 2016), in addition to machine translation by learning from multiple source and/or target languages (Dong et al.,

2015; Zoph and Knight, 2016; Johnson et al., 2017). Furthermore, some models have added supervision at both the word-level and sentence-level (Hashimoto et al., 2017; Rei and Søgaard, 2019); however less attention has been directed to combining **document-level** supervision with more fine-grained units (i.e., word or sentence supervision) (Cummins and Rei, 2018).

In this thesis, my main model leverages MTL and applies supervision at both word and document levels. To my knowledge, this is the first work to use MTL for coherence evaluation (i.e., predicting a coherence score for a document). Nonetheless, I note that Jernite et al. (2017) used MTL to learn 3 discourse-relevant tasks simultaneously: (1) whether a pair of sentences are in the correct order, (2) the type of coherence relation between two sentences and (3) given a sequence of three sentences and five candidate sentences from the same paragraph, which candidate comes after the sequence. They evaluated their model intrinsically (i.e., whether the tasks help each other) and extrinsically on paraphrase detection, subjectivity evaluation and question classification. However, they did not employ their model in coherence modeling or its different applications.

2.6 Model interpretability

Despite the high performance achieved by neural models on various problems, they are typically viewed as a ‘black box’ and it is not clear what linguistic features they capture. This is a common problem for deep learning models that has motivated a large body of work to focus on their interpretability (Sundararajan et al., 2017; Lundberg and Lee, 2017; Feng et al., 2018; Belinkov and Bisk, 2018). In this section, I give a brief background about deep learning interpretability methods, some of which I use in this thesis.

Intrinsic evaluation A common way to understand neural models is to inspect the quality of their learned representations via intrinsic evaluation. This approach has become ubiquitous with the spread of distributional semantic models. Investigating these semantic spaces can be achieved using various simple methods such as (1) comparing word vectors (e.g., measuring cosine similarity between vectors) to detect the relations between their respective words and (2) exploiting dimensionality reduction techniques (e.g., t-distributed Stochastic Neighbor Embedding (t-SNE; van der Maaten and Hinton, 2008)) to map the high-dimensional vectors to two dimensions and reveal the linguistic regularities in their embedding space (Collobert et al., 2011; Erk, 2012; Mikolov et al., 2013c,b; Levy and Goldberg, 2014; Faruqui et al., 2015; Farag, 2016).

Parameters of neural networks have also been visualised to understand the impact of input words on network output. One approach calculates the derivative of the final output with respect to input word embeddings in order to measure the contribution of input

words in the final network decision (Li et al., 2016a). A saliency heatmap is created in turn for visualisation, where more influential words are expected to have higher gradient norms. This approach was used to interpret a few sentence ordering models (Chen et al., 2016; Logeswaran et al., 2018). As for the models that employ attention mechanisms, a straightforward interpretability approach is to visualise the attention weights (a_t in Eq. 2.8) to examine which linguistic units the model focuses on (Bahdanau et al., 2015; Yang et al., 2016b; Lin et al., 2017; Chen et al., 2017; Bao et al., 2018; Wang and Wan, 2019). I use attention visualisation in this thesis. I note, however, that while attention could provide useful insights about what the models focus on, it should not be regarded as the only explanation for, or a fail-safe predictor of, model output (Serrano and Smith, 2019; Wiegrefe and Pinter, 2019).

Other work has taken a deeper look into model representations and inspected the functionality of certain dimensions of input and hidden activation vectors (Li et al., 2016b; Karpathy et al., 2016; Kádár et al., 2017).

Adversarial evaluation The idea behind adversarial evaluation is to apply small changes to the input examples with the intention of tricking the model into generating incorrect predictions. This type of evaluation reveals model vulnerabilities and thus helps us understand how the model works and how it could be further improved. Testing the robustness of models against adversarial examples has been widely investigated in the literature (Goodfellow et al., 2015; Jia and Liang, 2017; Mudrakarta et al., 2018; Belinkov and Bisk, 2018; Shi et al., 2018). Adversarial evaluation is often categorised into *white-box* examples that leverage knowledge about the model parameters (Goodfellow et al., 2015; Ebrahimi et al., 2018) or *black-box* examples that do not have explicit access to these parameters; I use the latter in this thesis. Several studies have exploited *black-box* adversarial evaluation. For example, Jia and Liang (2017) evaluated reading comprehension systems by inserting distracting sentences into the paragraphs that contain the answers, and showed that state-of-the-art models are vulnerable to these adversarial examples. Hosseini et al. (2017) showed that minor perturbations such as misspelling a word or adding a dot in between its characters in an input sentence drastically lower the accuracy of Google’s **Perspective** API for toxic comments detection. Belinkov and Bisk (2018) demonstrated that machine translation models could be deceived with natural and artificial kinds of noise, and Shi et al. (2018) attacked image captioning frameworks by replacing certain words in the captions. In this thesis, I use adversarial evaluation to examine the vulnerability of coherence models to syntactic and semantic changes. I also use adversarial evaluation to test the ability of AES systems to detect inputs of grammatical but incoherent sequences of sentences.

2.7 Evaluation

In this section, I present the main evaluation metrics used throughout this thesis.

Pairwise Ranking Accuracy (PRA) Pairwise Ranking Accuracy is the standard evaluation metric (Barzilay and Lee, 2004; Barzilay and Lapata, 2005) to evaluate coherence on binary synthetic datasets that consists of well-organised coherent documents and their incoherent counterparts created by permuting the sentence order in the coherent documents. PRA calculates the fraction of correct pairwise rankings in the test data; i.e., an original text should be ranked higher than its noisy counterparts.

Total Pairwise Ranking Accuracy (TPRA) Total Pairwise Ranking Accuracy is a more generalised metric for coherence evaluation; it ranks each original article against all the incoherent articles in the dataset, and not just its own permuted counterparts. TPRA was first used by Smith et al. (2016a) and I further motivate it in Farag et al. (2018).

Quadratic Weighted Kappa (QWK) Quadratic Weighted Kappa measures the agreement between two ratings. QWK ranges from -1 to 1 based on the degree of agreement (0 represents random agreement, negative values indicate agreement worse than chance and 1 is complete agreement). Using weighted Kappa differs from Cohen’s Kappa (Cohen, 1960) in that it accounts for the degree of disagreement and therefore, better represents ordinal classes. In order to estimate QWK between ratings by annotator₁ and ratings by annotator₂, 3 matrices are calculated as follows. First, a weight matrix $W \in \mathbb{R}^{N \times N}$ is computed by:

$$W_{i,j} = \frac{(i-j)^2}{(N-1)^2} \quad (2.13)$$

where i is the label given by annotator₁, j is the label given by annotator₂ and N is the number of possible ratings. Second, a histogram matrix is built $O \in \mathbb{R}^{N \times N}$ such that $O_{i,j}$ corresponds to the number of examples that are assigned label i by annotator₁ and label j by annotator₂. A histogram vector is composed for annotator₁’s ratings and another histogram vector is composed for annotator₂’s ratings where a vector element at position k indicates the number of examples that received a rating k . A third histogram matrix of expected ratings $E \in \mathbb{R}^{N \times N}$ is then calculated as the outer product between the two histogram vectors; E is normalized to have the same sum as O by simply dividing both matrices by their sum so that they each have a sum of one. Finally, using the three matrices, QWK is calculated as:

$$qwk = 1 - \frac{\sum_{i,j} W_{i,j} O_{i,j}}{\sum_{i,j} W_{i,j} E_{i,j}} \quad (2.14)$$

QWK typically measures the agreement between two annotators. In order to adapt QWK for more than two annotators, *leave-one-out resampling* (Weiss and Kulikowski, 1991) could be used, where for each annotated example one annotator’s label is randomly chosen to be rating_1 and the mean of the other annotators’ labels is rating_2 ; QWK is then calculated between rating_1 and rating_2 . This process is repeated for M times and the final QWK is estimated as the mean of all the calculated QWK values. QWK has been used as a standard evaluation metric for essay scoring systems where annotator_1 represents the tested AES system and annotator_2 represents the human grader (Phandi et al., 2015; Taghipour and Ng, 2016; Dong et al., 2017; Jin et al., 2018).

Pearson’s product-moment correlation coefficient (r) Pearson’s r is a parametric measure that determines the strength of linear dependence between two variables. It attempts to fit the data points of the two variables into a single line, and therefore, is sensitive to outlier points. The value of $r \in [-1, 1]$, where $r = 1$ denotes a total positive linear correlation, $r = -1$ denotes a total negative linear correlation, and $r = 0$ means no correlation.

Spearman’s rank correlation coefficient (ρ) Spearman’s ρ is a non-parametric measure that determines the degree of association between two ranked variables. In contrast to Pearson’s r , Spearman’s ρ measures the monotonic relationship between the two variables, not the linear relationship, and is only dependent on the ordinal arrangement of the variables, and therefore, is not sensitive to outlier points. The value of $\rho \in [-1, 1]$, where $\rho = 1$ denotes a perfect positive association, $\rho = -1$ denotes a perfect negative association, and $\rho = 0$ means no association. Pearson’s and Spearman’s correlations have also been used to evaluate AES systems (Yannakoudakis et al., 2011; Yannakoudakis and Cummins, 2015; Alikaniotis et al., 2016; Dasgupta et al., 2018).

Kendall rank correlation coefficient (Kendall’s Tau (τ)) Kendall’s τ is also a non-parametric rank correlation used to measure the ordinal association between two variables. The value of $\tau \in [-1, 1]$, where $\tau = 1$ denotes a perfect positive association, $\tau = -1$ denotes a perfect negative association, and $\tau = 0$ means no association. Kendall’s τ determines the strength of association based on the concordance and discordance between pairs of observations, where the pair (x_i, x_j) and (y_i, y_j) is concordant if $x_i - x_j$ and $y_i - y_j$ have the same sign and discordant otherwise. Accordingly, Kendall’s τ has been used in information ordering tasks to measure the degree of similarity between the sentence order in a permuted document and its original version (Lapata, 2006). It is more suitable than Spearman’s correlation in measuring this similarity as it is more accurate when the sample size is small (Kendall and Gibbons, 1990; Lapata, 2006). Furthermore, Lapata (2006) have shown that Kendall’s τ ranks correlate with human judgement of overall

text understandability and coherence. In order to calculate Kendall’s τ , let π and σ be the orderings of the original and permuted article respectively and $S(\pi, \sigma)$ be the minimum number of adjacent transpositions needed to transform σ back to π ; Kendall’s τ is calculated by:

$$\tau = 1 - \frac{2S(\pi, \sigma)}{N(N-1)/2} \quad (2.15)$$

where N is the number of sentences.

Fisher transformation Some of the tasks presented in this thesis consist of multiple datasets (e.g., the essay scoring task in Chapter 6 has 8 prompts), and are evaluated by calculating kappa or correlation values for each dataset separately. Nonetheless, aggregating the performance across all these datasets is useful to give an overall indication of model performance on the task. Simply averaging the kappa/correlation values may not be accurate as their sampling distribution might be skewed (Silver and Dunlap, 1987). Therefore, to remedy this, Fisher transformation is applied as it is approximately a variance-stabilizing transformation.⁹ Fisher transformation to a kappa/correlation value (v) is defined as:

$$z = \frac{1}{2} \ln \frac{1+v}{1-v} \quad (2.16)$$

The mean of all the transformed kappa/correlation values (z) is then calculated (I denote this mean by \bar{z}) and the final average kappa/correlation (v_{avg}) that measures the overall performance is calculated by applying the reverse transformation to \bar{z} :

$$v_{avg} = \frac{e^{2\bar{z}} - 1}{e^{2\bar{z}} + 1} \quad (2.17)$$

F1 score F1 score is a measurement of accuracy based on the harmonic mean of precision and recall:

$$F1 = \frac{2 \times \textit{precision} \times \textit{recall}}{\textit{precision} + \textit{recall}} \quad (2.18)$$

where precision is calculated as:

$$p = \frac{\textit{true positives}}{\textit{true positives} + \textit{false positives}} \quad (2.19)$$

and recall is calculated as:

$$r = \frac{\textit{true positives}}{\textit{true positives} + \textit{false negatives}} \quad (2.20)$$

Accordingly, the F1 score gives equal emphasis to both precision and recall.

⁹It was recommended to use Fisher transformation before averaging QWK scores for the AES contest by Kaggle (<https://www.kaggle.com/c/asap-aes/overview/evaluation>).

APPROACH

In this chapter, I present my neural discriminative approach to coherence modeling that aims to capture both local and global aspects of coherence. Concretely, I propose a hierarchical neural network trained in a multi-task learning (MTL) fashion that learns to predict a document-level coherence score (at the top layer of the network) together with word-level syntactic information (at lower layers), taking advantage of the hierarchical inductive transfer between the two tasks. The syntactic information is either grammatical roles (GRs) or part-of-speech (POS) tags. Additionally, I extend my approach by integrating contextualised word embeddings, in particular ELMo and BERT embeddings. I start, in §3.1, by describing my basic hierarchical model that performs the single task of predicting a document-level coherence score and describe the neural representations generated at each network level. I then discuss, in §3.2, my MTL framework, motivate the auxiliary functions I optimise (i.e., predicting word-level GRs or POS tags) and explain their relevance to coherence assessment. Finally, in §3.3, I present models that leverage syntactic labels in different fashions to further validate my MTL approach, and in §3.4, I summarise the chapter.

3.1 Single-task learning model

In this section, I describe my baseline model that performs the single task of predicting an overall coherence score for a given document. The single-task learning (STL) model is lexical; it only leverages input word representations retrieved from a pre-trained space. The model is later extended with syntactic information either via feeding this information as input to the model (§3.3), or allowing the model to learn it in a multi-task fashion (§3.2). The STL model is a hierarchical Bi-LSTM-based neural network. Hierarchical networks are an attractive approach to encode the structure of a document (Li et al., 2015; Yang et al., 2016b), where a document is composed of a sequence of sentences $\{s_1, \dots, s_i, \dots, s_m\}$

and, in turn, each sentence consists of a sequence of words $\{w_1, \dots, w_t, \dots, w_n\}$. In some cases, the document might contain paragraphs $\{p_1, \dots, p_j, \dots, p_l\}$, adding another level to the structure of the document. Accordingly, a document representation is built in a bottom-up fashion:

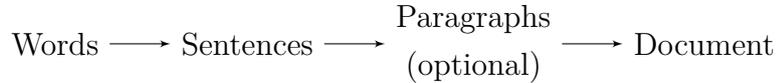


Figure 3.1: Document Structure

The hierarchical architecture we use follows the work of Yang et al. (2016b); the core difference is that our model is LSTM-based and theirs is GRU-based. Our main contribution is in our MTL framework (§3.2). The following subsections detail how each level in the network hierarchy is encoded and Fig. 3.2 graphically illustrates the network architecture (excluding the red-dotted box that is specific to MTL).

3.1.1 Word representation

The first step to encode a document is to initialise its word representations; I use one of two representation types: standard and contextualised.

Standard embeddings In the standard embeddings setup, a vocabulary V is constructed from the words that occur in the training data wherein each word has a unique index $k \in [1, |V|]$. A lookup table (i.e., word embedding matrix $E \in \mathbb{R}^{|V| \times d^w}$) is then constructed, where the k -th row corresponds to the feature vector of the k -th word in V and d^w is the vector length. The word vectors are retrieved from a pre-trained embedding space (§2.3.2).

In order to process an input document, each sentence (s_i) is translated into a sequence of its word indices in V , then fed to the neural network. The first layer in the network performs a lookup operation in E to obtain the semantic representations of words. For instance, if

$$E = \begin{bmatrix} [vector_1] \\ \vdots \\ [vector_{|V|}] \end{bmatrix}$$

and the input document consists of two sentences, each contains three words mapped to their

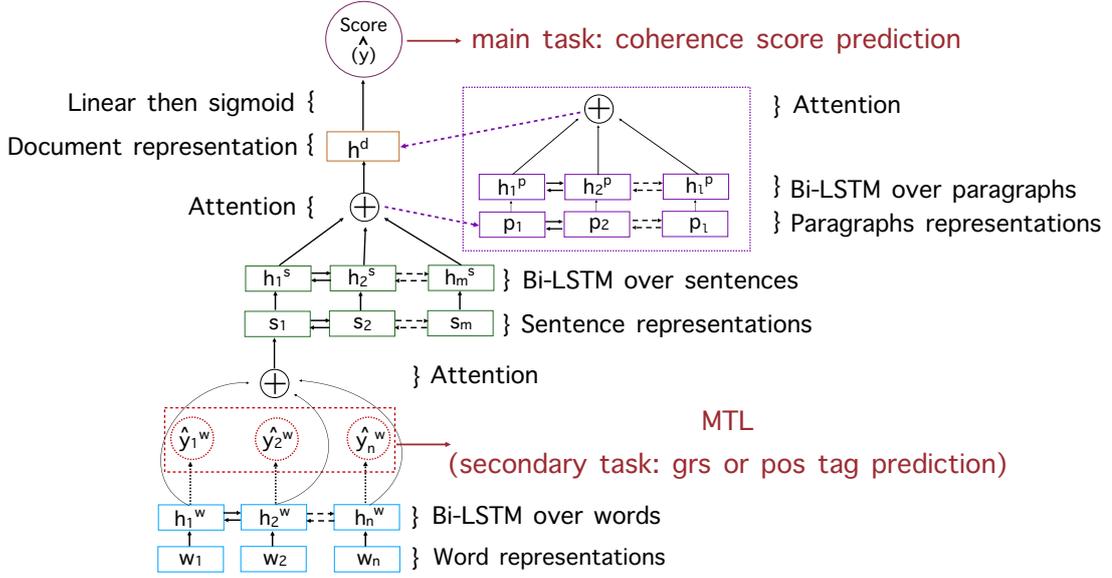


Figure 3.2: The architecture of the STL and MTL models. The dotted red box is specific to the MTL framework. The purple box is applied if the document contains paragraph boundaries (which is the case for the Grammarly Corpus in §4.1.2) in order to create paragraph representations prior to the document one.

indices,¹ (e.g., $[[6, 14, 2], [7, 90, 2]]$), the document will be translated into a matrix $\in \mathbb{R}^{2 \times 3 \times d^w}$:

$$\left[\begin{array}{c} \left[\begin{array}{c} [vector_6] \\ [vector_{14}] \\ [vector_2] \end{array} \right], \left[\begin{array}{c} [vector_7] \\ [vector_{90}] \\ [vector_2] \end{array} \right] \end{array} \right]$$

If there are paragraphs, there will be a fourth dimension to represent the number of paragraphs.

Contextualised embeddings In another setup, I bootstrap the network with contextualised word vectors. In contrast to standard embeddings, the representation of each word is a function of the entire sentence it appears in. I evaluate my models with two types of pre-trained vectors: ELMo (Bi-LSTM-based) and BERT (transformer-based); see §2.3.2 for further details about the two models. I use a feature-based approach where I extract the contextualised representations from their pre-trained models and use them to initialise my network. For ELMo, I use two approaches to extract the representations:

¹I presume that sentences are of the same length for simplicity. In practice, sentences are of different lengths and this difference is addressed by padding as will be discussed in §4.3.

1. Only take the top layer in the three-layer representation (Peters et al., 2017).
2. Average the three layers to form a single vector (Peters et al., 2018).

As for BERT, I leverage BERT_{LARGE} (cased) and specifically use layer 16 and represent each word by the average of its subword representations, following previous work (Hewitt and Manning, 2019). This choice is also supported by other studies that have shown that syntactic information is better captured in the middle layers (Jawahar et al., 2019), semantic features are spread across all the layers (Tenney et al., 2019a) and the middle layers are the most transferable (Liu et al., 2019b).² I motivate employing contextualised embeddings in my work as follows:

- Learning representations for words based on their context lends itself to word sense disambiguation (WSD) and thus builds semantically rich sentence embeddings which is key to capturing connections and interactions between sentences.
- Contextualised embeddings are capable of encoding semantic and syntactic features which is shown by their performance on relevant tasks such as POS and semantic tagging and coreference resolution (Peters et al., 2018; Tenney et al., 2019b,a; Liu et al., 2019b). Capturing such features is useful for coherence modeling, as will be discussed in §3.2.1 and §3.2.2.
- Contextualised embeddings allow me to compare the results of initialising the STL network with these embeddings that carry semantic and syntactic information against learning this information via MTL. I also take the comparison further and integrate ELMo and BERT in the MTL-based models in order to verify whether MTL can capture different linguistic properties than the ones encapsulated in contextualized embeddings.

With the different word initialisation methods, I end up with 4 versions of the STL model: STL (using standard embeddings), STL+ELMo (using ELMo top layer), STL+ELMo-avg (using the average of ELMo layers) and STL+BERT. Contextualised embeddings are also integrated in the MTL framework as will be discussed in §3.2.

3.1.2 Sentence representation

Since I model coherence as transitions and interactions between sentences, building high quality sentence representations is important for the task. As discussed in §2.3.1 and §2.3.3, LSTM is a popular strategy to encode sentences and it has been utilised in numerous coherence assessment models (§2.4). I employ a Bi-LSTM on the sequence of word

²See the per layer performance on a variety of linguistic tasks in tables 6, 8 and 10 in Liu et al. (2019b).

vectors in each sentence to construct richer representations that encode information from the forward and backward directions, and then concatenate the output vectors of both directions:³

$$\begin{aligned}\overrightarrow{h}_{it}^w &= LSTM(w_{it}, \overrightarrow{h}_{it-1}^w) \\ \overleftarrow{h}_{it}^w &= LSTM(w_{it}, \overleftarrow{h}_{it+1}^w) \\ h_{it}^w &= [\overrightarrow{h}_{it}^w, \overleftarrow{h}_{it}^w]\end{aligned}\tag{3.1}$$

where w_{it} is the input word representation and $h_{it}^w \in \mathbb{R}^{dim^w}$, where dim^w is a hyperparameter indicating the hidden layer size.⁴

Subsequently, a sentence representation is composed by applying an attention mechanism to aggregate the output hidden states (§2.3.3):

$$\begin{aligned}u_{it}^w &= \tanh(W^w h_{it}^w) \\ a_{it}^w &= \frac{\exp(v^w u_{it}^w)}{\sum_t \exp(v^w u_{it}^w)} \\ s_i &= \sum_t a_{it}^w h_{it}^w\end{aligned}\tag{3.2}$$

where W^w and v^w are learnable parameters. Attention allows the model to focus on the salient words for coherence and build better sentence representations.

3.1.3 Paragraph representation

The final document-level representation could be inferred directly from its composing sentences, in case the document consists of one paragraph or we want to ignore paragraph boundaries. However, if we want to model paragraphs in multi-paragraph documents, then the respective representations should be constructed. A paragraph is a coherent unit of text focusing on a specific topic, while transitioning to a new paragraph usually signals topic shift. Building paragraph representations from sentences is similar to building sentence representations from words:

$$\begin{aligned}\overrightarrow{h}_{ji}^s &= LSTM(s_{ji}, \overrightarrow{h}_{ji-1}^s) \\ \overleftarrow{h}_{ji}^s &= LSTM(s_{ji}, \overleftarrow{h}_{ji+1}^s) \\ h_{ji}^s &= [\overrightarrow{h}_{ji}^s, \overleftarrow{h}_{ji}^s]\end{aligned}\tag{3.3}$$

³I use the notations w and s to refer to word and sentence representations respectively, and subscripts t and i to denote the indices of words and sentences respectively. i.e. w_{it} is the t^{th} word in the i^{th} sentence and s_i is the i^{th} sentence in the document. In case paragraphs exist, p is used to indicate paragraph representations and a subscript j is added to the word/sentence notations to refer to the index of their encompassing paragraph.

⁴For more detailed LSTM equations, see Eq. 2.2.

where s_{ji} is the i^{th} sentence in the j^{th} paragraph computed in Eq. 3.2 and $h_{ji}^s \in \mathbb{R}^{dim^s}$. Attention is then applied to aggregate the vectors generated by the Bi-LSTM:

$$\begin{aligned} u_{ji}^s &= \tanh(W^s h_{ji}^s) \\ a_{ji}^s &= \frac{\exp(v^s u_{ji}^s)}{\sum_i \exp(v^s u_{ji}^s)} \\ p_j &= \sum_i a_{ji}^s h_{ji}^s \end{aligned} \quad (3.4)$$

where W^s and v^s are learnable weights.

3.1.4 Document representation

In order to build the final document vector (h^d) that is to be scored, a Bi-LSTM is applied to either the sentence representations in Eq. 3.2 (denoting a single-paragraph text) or the paragraph representations in Eq. 3.4 (denoting a multi-paragraph text):

$$\begin{aligned} \vec{h}_i^s &= LSTM(s_i, \vec{h}_{i-1}^s) \quad (or) \quad \vec{h}_j^p = LSTM(p_j, \vec{h}_{j-1}^p) \\ \overleftarrow{h}_i^s &= LSTM(s_i, \overleftarrow{h}_{i+1}^s) \quad (or) \quad \overleftarrow{h}_j^p = LSTM(p_j, \overleftarrow{h}_{j+1}^p) \\ h_i^s &= [\vec{h}_i^s, \overleftarrow{h}_i^s] \quad (or) \quad h_j^p = [\vec{h}_j^p, \overleftarrow{h}_j^p] \end{aligned} \quad (3.5)$$

where $h_j^p \in \mathbb{R}^{dim^p}$. Similar to the previous steps, an attention function is applied over the hidden states of sentences or paragraphs:

$$\begin{aligned} u_i^s &= \tanh(W^s h_i^s) \quad (or) \quad u_j^p = \tanh(W^p h_j^p) \\ a_i^s &= \frac{\exp(v^s u_i^s)}{\sum_i \exp(v^s u_i^s)} \quad (or) \quad a_j^p = \frac{\exp(v^p u_j^p)}{\sum_i \exp(v^p u_j^p)} \\ h^d &= \sum_i a_i^s h_i^s \quad (or) \quad h^d = \sum_j a_j^p h_j^p \end{aligned} \quad (3.6)$$

where $h^d \in \mathbb{R}^{dim^d}$. The resulting document vector (h^d) encompasses useful features learnt at the different levels of the network. Applying a Bi-LSTM at each level facilitates learning contextually rich representations and following that by attention is a selection strategy to only forward the salient information to the next layer. Furthermore, my hierarchical network is an architecture designed to be able in principle to capture aspects of local coherence by using sentence-level Bi-LSTM which encodes each sentence in its local context of preceding and successive sentences. Nonetheless, it could further capture global coherence by constructing a document vector via attending to all its sentences/paragraphs. In addition, modeling the interactions between paragraphs, in multi-paragraph texts, could detect topic shifts, further promoting the global coherence of discourse.

3.1.5 Scoring

In order to predict the overall coherence score for text, a linear transformation is applied to the document vector (h^d) followed by a sigmoid operation to bound the score in $[0, 1]$:

$$\hat{y} = \sigma(W^d h^d) \quad (3.7)$$

where W^d is the linear function weight. The scoring process varies according to the dataset and its number of labels as follows:

1. In binary datasets, documents are labeled as either coherent ($y = 1$) or incoherent ($y = 0$). The network then predicts one score and attempts to push it towards 1 or 0 based on the true label. In that case, in Eq. 3.7, $W^d \in \mathbb{R}^{dim^d}$ where dim^d represents the dimensionality of the document vector. The parameters of the network are optimised to minimise the negative log-likelihood of the ground-truth label y , given the predicted score \hat{y} , and thus the main loss is calculated by:

$$L_{main} = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}) \quad (3.8)$$

2. In multi-class datasets, documents are labeled with one of different degrees of coherence: $y \in C$ where $|C| > 2$. Specifically, each document is labeled with a one-hot vector with length $|C|$ with a value 1 at the index of the correct class and 0 everywhere else. Accordingly, the model predicts $|C|$ scores, using Eq. 3.7 with $W^d \in \mathbb{R}^{|C| \times dim^d}$, and learns to maximise the value corresponding to the gold label. For optimisation, I use Mean Squared Error (MSE) to minimise the discrepancy between the one-hot gold vector and the estimated one:

$$L_{main} = \frac{1}{|C|} \sum_{j=1}^{|C|} (y_j - \hat{y}_j)^2 \quad (3.9)$$

The final prediction of the model is the class with the highest predicted score. An alternative approach to the multi-class problem is to apply a softmax over the predictions instead of a sigmoid (Eq. 3.7), and minimise the categorical cross entropy; however, initial experiments on the development set showed that my formulation yields better results.

3.2 Multi-task learning model

The model described in 3.1 performs the single task of predicting a coherence score for a text, and hence all model parameters are tuned to minimise the main loss (L_{main} in Eqs. 3.8

and 3.9). I extend this model to a multi-task learning (MTL) framework by training it to optimise a secondary objective at the bottom layers of the network, along with the main one, as shown in Fig. 3.2 (red dotted-box). Specifically, the model is trained to predict a document-level coherence score along with word-level labels indicating syntactic properties of words.⁵ I experiment with two types of word-level labels: grammatical roles (GRs) and part-of-speech (POS) tags. The choice of these labels in particular is motivated by their relevance to coherence assessment as will be discussed in the following subsections. I present the first MTL model that learns to predict these syntactic labels as an auxiliary task in order to enhance the performance on the task of coherence assessment. Learning both tasks in a hierarchical network allows us to take advantage of hierarchical inductive transfer between them and learn linguistically rich representations at the bottom layers that can be exploited by the top layers of the network. Further details about these syntactic labels are provided in §3.2.1 and §3.2.2, but first I explain how the secondary objective is integrated to the network described in §3.1.

In order to create the secondary labels, each word in the input document is labeled with a syntactic class in Y^w which is a pre-defined set of all possible labels generated by parsing the training data. More specifically, each word is labeled with a one-hot vector of size $|Y^w|$ with value 1 at the the index of its ground-truth class and 0 everywhere else. Subsequently, the model is trained to predict a probability distribution over Y^w for each input word by applying a linear operation normalized by a softmax function over the word representation h_{it}^w (or h_t^w for simplicity) from Eq. 3.1 as follows. First, the model calculates the dot product of a learned weight matrix $W^a \in \mathbb{R}^{|Y^w| \times \dim^w}$ (\dim^w is defined in §3.1.2) and each word representation h_t^w to get a vector of logit scores for each word, in which each element represents the score assigned to each class in Y^w . These predicted scores are, however, unnormalized log probabilities which leads to the second step of applying a softmax function to turn the scores into a probability distribution over Y^w . More formally, the probability distribution over all the classes in Y^w for the word at position t is calculated by:

$$P(y_t^w = r | h_t^w) = \frac{\exp(W_r^a h_t^w)}{\sum_{r' \in Y^w} \exp(W_{r'}^a h_t^w)} \quad (3.10)$$

where subscript r in W_r^a refers to the r^{th} row in W^a , and $0 \leq r < |Y^w|$. Adding supervision for syntactic labels at h_t^w follows the previous work of Søgaard and Goldberg (2016). However, in contrast to their work that focuses on word-level tasks (e.g., POS tagging and syntactic chunking), I combine word-level with document-level tasks.

Equation 3.10 is applied for all the labels ($r \in Y^w$) which results in a vector of probabilities over Y^w . The network aims at maximising the probability corresponding to

⁵My code for the MTL model is available at https://github.com/Youmna-H/coherence_mtl.

the true label and hence optimises an auxiliary objective of categorical cross-entropy; i.e., the negative log-probability of the correct labels, for all the words in the document:

$$L_{aux} = - \sum_t \sum_r y_r^w \log P(y_t^w = r | h_t^w) \quad (3.11)$$

where y_r^w either equals 1 if it is the correct class for the word or 0 otherwise. Both the main loss (Eqs. 3.8 and 3.9) and the auxiliary one (Eq. 3.11) are optimised jointly, but with different weights indicating the importance of each task:

$$L_{total} = \alpha L_{main} + \beta L_{aux} \quad (3.12)$$

where $\alpha, \beta \in [0, 1]$ are loss weight hyperparameters.⁶ The advantages of learning syntactic properties in an MTL framework, in comparison to feeding them as input features, include:

- Efficiency, as using parsers to extract syntactic labels is limited to training data. Additionally, predicting these labels is only required during training and therefore, at inference time, MTL uses the same number of parameters as STL.
- The ability to control how much the model needs to learn from each task, by tuning α and β in Eq. 3.12.

The following subsections explain the syntactic information I use as labels for the auxiliary task.

3.2.1 Multi-task learning with grammatical roles

Grammatical roles (GRs) (alternatively called grammatical relations or grammatical functions) refer to syntactic roles taken by words in a sentence based on their relations to other words in the same sentence, such as *subject*, *direct object*, *determiner*, etc. These roles are defined within the grammatical dependency structure of sentences, where a sentence has a head (root) on which every other word depends directly or indirectly.⁷ Dependency structures are often extracted using dependency parsers (Chen and Manning, 2014; Dozat and Manning, 2017) and traditionally, the main verb takes the role of the sentence root. A grammatical relation consists of three participants: a head, a dependent and the type/label of the relation between the head and the dependent as illustrated in Fig. 3.3. A GR is the type of the relation and is assigned to the dependent word. Accordingly, each word in my training data is annotated with a GR that denotes the type of the relation the word participates in as a dependent, and the root is labeled with a *root* label. For instance, the

⁶I note that Eqs. 3.8, 3.9 and 3.11 calculate the loss for one document. However, the mean loss for the documents in each training mini-batch is calculated and optimised, which I remove from the equations for simplicity.

⁷Dependency grammar is often traced back to the work of Tesnière (1959).

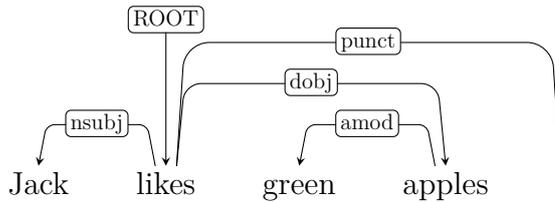


Figure 3.3: A grammatical dependency structure generated by the Stanford Dependency Parser (v. 3.8) (Chen and Manning, 2014). Each grammatical relation is represented with an arrow where the head of the relation is the start of the arrow, the dependent is its end and the relation type is written in the box. In this graph, *nsubj* is *nominal subject*, *dobj* is *direct object*, *amod* is *adjectival modifier* and *punct* is *punctuation*.

sentence in Fig. 3.3 is labeled with the sequence: “*nsubj, root, amod, dobj, punct*”. As GR annotation schemes are different across languages, the Universal Dependencies (UD) project has been developed to create cross-linguistically consistent annotations for many languages (Nivre et al., 2016), which I utilise in this thesis. The full list of GRs is detailed in Table 3.1.

Learning the GRs of words is a way to capture sentence semantics which is useful for many NLP tasks such as question answering (Hakimov et al., 2013), summarisation (Sakhare and Kumar, 2014) and machine translation (Sennrich and Haddow, 2016). GRs have also been exploited for coherence modeling, more commonly with entity-based approaches (Barzilay and Lapata, 2008; Elsner and Charniak, 2011b; Guinaudeau and Strube, 2013; Tien Nguyen and Joty, 2017; Joty et al., 2018), due to their relevance to the notion of salience (§2.1.4); they were also used in other probabilistic coherence models (Lapata, 2003). Furthermore, GRs were leveraged in abstractive summarisation (Fang and Teufel, 2014) as an implementation of Kintsch and Van Dijk’s 1978 model of human comprehension that describes how a text is represented in a reader’s memory, and which textual elements are salient and thus will be recalled later as the reader processes the text.

Inspired by previous work, I propose an MTL framework that leverages the hierarchical inductive bias between learning GRs and coherence assessment by predicting word-level GRs as a secondary objective together with the main document-level coherence scoring one. Similar to STL, I create different versions of the network where I use standard or contextualised embeddings: MTL_{GRs} , $MTL_{GRs}+ELMo$ and $MTL_{GRs}+BERT$.⁸ In this MTL setup, the labels in Y^w are the GRs extracted by a dependency parser (the left half of Table 3.1), and the network learns to assign the correct GR to each input word. Contrary to entity-based approaches, I leverage all the GR types in Table 3.1, and predict them for all the words in the input document, not just subject and object roles for entities. The motivation for using the full set of GRs is two-fold:

⁸From STL experiments, I found that using the top ELMo layer outperforms averaging the layers, and hence, I apply the first in the rest of the experiments.

GR Type	Description	POS Tag	Description
ACL [relcl]	clausal modifier of noun (adjectival clause)	CC	coordinating conjunction
ADVCL	adverbial clause modifier	CD	cardinal number
ADVMOD	adverbial modifier	DT (or DET)	Determiner
AMOD	adjectival modifier	EX	Existential <i>there</i>
APPOS	appositional modifier	FW	foreign word
AUX	auxiliary	IN	preposition or subordinating conjunction
AUXPASS	passive auxiliary	JJ	adjective
CASE	case marking	JJR	adjective comparative
CC [preconj]	coordinating conjunction	JJS	adjective superlative
CCOMP	clausal complement	LS	list item marker
COMPOUND [prt]	compound	MD	modal
CONJ	conjunct	NN	noun, singular or mass
COP	copula	NNS	noun plural
CSUBJ	clausal subject	NNP	proper noun, singular
CSUBJPASS	clausal passive subject	NNPS	proper noun, plural
DEP	unspecified dependency	PDT	Predeterminer
DET [predet]	determiner	POS	possessive ending
DISCOURSE	discourse element	PRP	personal pronoun
DOBJ	direct object	PRP\$	possessive pronoun
EXPL	expletive	RB	adverb
IOBJ	indirect object	RBR	adverb, comparative
MARK	marker	RBS	adverb, superlative
MWE	multi-word expression	RP	particle
NEG	negation modifier	SYM	symbol
NMOD [tmod, poss, npmod]	nominal modifier	TO	to
NSUBJ	nominal subject	UH	interjection
NSUBJPASS	passive nominal subject	VB	verb, base form
NUMMOD	numeric modifier	VBD	verb, past tense
PARATAXIS	parataxis	VBG	verb, gerund or present participle
PUNCT	punctuation	VBN	verb, past participle
ROOT	root	VBP	verb, non-3rd person singular present
XCOMP	open clausal complement	VBZ	verb, 3rd person singular present
		WDT	wh-determiner
		WP	wh-pronoun
		WP\$	possessive wh-pronoun
		,	,
		.	.
		“	“
		”	”
		:	:
		\$	\$

Table 3.1: The left half displays the GRs (based on the UD scheme) extracted from the WSJ training data (§4.1.1), the same roles are extracted from the Grammarly Corpus of Discourse Coherence GCDC (§4.1.2). The text inside the square brackets in the leftmost column denotes the extracted subtypes (language specific types). For more details about subtypes, see <http://universaldependencies.org/docsv1/ext-dep-index.html>. The total number of main types and their subtypes is 39. For the full list of UD, see <http://universaldependencies.org/docsv1/u/dep/index.html>. The right half lists the POS tags and their description, following the Penn Treebank project (<https://catalog.ldc.upenn.edu/docs/LDC99T42/tagguid1.pdf>). The total number of POS tags is 41.

1. Using all GRs enhances the prediction of subject and object roles for entities by learning these roles in their full context (as will demonstrated in §4.4.3.2).
2. Utilising all the words in the input sequence, not just the words denoting entities, allows the model to learn other coherence relevant features such as rhetorical relations that are not represented by entities alone. Words with other grammatical roles such as verbs are also important in building discourse relations (Lapata, 2003; Asher and Lascarides, 2003). Consider the following examples:

(a) I booked my train ticket to Paris. I am travelling tomorrow.

(b) I booked my train ticket to Paris. I am cooking tomorrow.

The verbs in the second sentence of both examples (*travelling* and *cooking*) play the key role in determining text coherence, making example (a) more coherent than (b). An EGrid model would not be able to distinguish between both examples. Furthermore, EGrid models require feature engineering at train and test time to build the EGrids.

3.2.2 Multi-task learning with part-of-speech tags

In this section, I investigate the use of another type of syntactic information in coherence modeling which is POS tags. Utilising the POS features of the words in a sentence helps capture its syntax, thus builds better sentence representations. POS features have been widely used in numerous NLP problems such as essay scoring (Yannakoudakis et al., 2011), machine translation (Niehues and Cho, 2017) and question answering (Hommel et al., 2019). They have also been leveraged in discourse related tasks such as detecting implicit coherence relations (Lin et al., 2009), in addition to coherence modeling by capturing the intentional structure of discourse (see Louis and Nenkova (2012) in §2.2.2).

Encouraged by the work of Louis and Nenkova (2012), I implement an MTL framework where I use word-level POS tag prediction as a secondary training objective, where the network learns to assign a POS category to each input word. I also create three versions of the model based on the initialisation method: MTL_{POS} , $MTL_{POS}+ELMo$ and $MTL_{POS}+BERT$. In this MTL setting, the labels in Y^w are the POS tags parsed by a POS tagger (see right half of Table 3.1) and each word in an input sentence is labeled with a tag in Y^w . For instance, the corresponding sequence of labels to the input sentence in Fig. 3.3 is “*NNP VBZ JJ NNS .*” My motivation for using POS tags is:

- to facilitate capturing syntactic patterns and thereby model the intentional structure of text as shown by Louis and Nenkova (2012).
- to compare the impact of using different syntactic features (POS tags vs. GRs) on coherence assessment.

3.3 Neural syntactic models

In order to investigate the value of adopting MTL for coherence modeling, I compare it to neural models that incorporate syntactic information in different ways. I examine whether leveraging syntactic properties by learning them via MTL has an advantage over feeding them as input to the network. I also compare different MTL setups by learning a subset of GRs or jointly learning GRs and POS tags. All the models in this section are initialised with standard embeddings and detailed as follows.

Concatenation models Instead of learning to predict syntactic features within an MTL framework, I incorporate them as input features to the model by concatenating them to the word representations in the STL framework. In this setup, I randomly initialise an embedding matrix $E_{concat} \in \mathbb{R}^{|Y^w| \times g}$, where g is the embedding size, Y^w , as defined in §3.2, is the set of syntactic features and each feature is mapped to a row in E_{concat} . In order to process an input document, each word vector ($w_1 \dots w_n$ in Fig. 3.2) is concatenated with its syntactic label vector from E_{concat} . Here, the syntactic features are needed as input at both training and test time, unlike in MTL, where they are only required during training. I refer to the model that leverages GRs as *concat_{GRs}* and the one that uses POS tags as *concat_{POS}*.

Multi-task learning with SOX (MTL_{sox}) As elaborated in 3.2.1, I utilise all the GR types extracted from the training data, contrary to previous entity-based approaches that only focus on subject and object roles of entities. Therefore, in order to further assess the impact of my extended set of GRs, I re-train the same MTL model but now only utilise subject (S) and object (O) types as my secondary training signal and map any ‘other’ role to ‘X’; specifically, $Y^w = \{S, O, X\}$. For instance, in this MTL_{sox} approach, the input sentence in Fig. 3.3 will be labeled with the sequence: “S, X, X, O, X”. Furthermore, any word parsed as *nominal subject* (*nsubj*) is labeled as subject, and any word parsed as *direct object* (*dobj*), *indirect object* (*iobj*) or *passive nominal subject* (*nsubjpass*) is labeled as object.

Multi-task learning with two auxiliaries (MTL_{GRs+POS}) I create another setting for MTL with two auxiliary losses: one for GRs and the other for POS tags. In this setup, there are two sets of word-level classes (Y_1^w and Y_2^w) representing the sets of GRs and POS tags and the model learns to predict both labels for each input word. Accordingly, Eq. 3.10 is applied twice with different W^a weights for each set of classes and similarly Eq. 3.11 is applied to obtain two auxiliary losses: L_{aux1} and L_{aux2} . The final total loss the network

optimises is a weighted sum of all the losses:

$$L_{total} = \alpha L_{main} + \beta_1 L_{aux1} + \beta_2 L_{aux2} \quad (3.13)$$

where α , β_1 and β_2 are hyperparameters to be tuned. The motivation for this setup is to examine whether there are any further gains from learning both GRs and POS tags and whether they can capture complementary features. I refer to this network as $MTL_{GRs+POS}$.

3.4 Summary

In this chapter, I have presented my MTL approach to coherence modeling. I have first explained the main hierarchical attention-based architecture that performs the single-task of predicting a document-level coherence score (the STL model). I have then showed how the model is modified to perform MTL, where it learns to predict a document-level coherence score at its top layers together with word-level syntactic labels at its lower layers. I investigated two types of syntactic labels: GRs, inspired by entity-based discourse approaches and models of human comprehension, and POS tags, inspired by the role syntactic constructions play in modeling the intentional structure of discourse. I have also created variants of my models enhanced with ELMo (Bi-LSTM-based contextualised embeddings) or BERT (transformer-based contextualised embeddings). Finally, as a further investigation to my MTL approach, I have incorporated syntactic labels to my hierarchical network in different fashions by: concatenating the labels with the input word representations in the STL network, learning a subset of the GRs in MTL by focusing only on subject and object roles, or learning both GRs and POS tags as auxiliary labels in the MTL framework.

EXPERIMENTS

In this chapter, I present my experiments using the models described in Chapter 3 and compare them to state-of-the-art coherence models. I specifically assess my approach on the standard binary discrimination task that ranks a coherent original document against its incoherent counterparts created by distorting the sentence order in the original text. Furthermore, I evaluate my approach on realistic data of everyday writing, such as emails and online posts, that exhibits various coherence levels. I compare my models to a wide variety of benchmarks and state-of-the-art models. My experiments show the effectiveness of MTL, particularly when enhanced with contextualised embeddings, on the binary task of ranking a coherent document higher than *its* noisy versions. Furthermore, MTL with contextualised embeddings attains state-of-the-art accuracy using a more comprehensive evaluation setting that ranks a coherent document higher than *all* the incoherent documents in the dataset, not just its incoherent counterparts. MTL also achieves state-of-the-art performance on the realistic domain that contains texts of varying degrees of coherence; however, with further investigation I find that it fails to capture medium levels of coherence. Finally, I provide further analysis and visualisation to the models in order to interpret their behaviour and explicate their obtained results. I note that this chapter is based on a long paper published in the 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019) (Frag and Yannakoudakis, 2019).

This chapter is structured as follows. In §4.1, I present the binary synthetic dataset and the realistic one that exhibits multiple levels of coherence. Next, in §4.2, I present previous neural models that I compare my approach to, including models that achieved state-of-the-art results in coherence modeling. After that, in §4.3, I explain how the models are trained and the hyperparameters they use. In §4.4 and §4.5, I discuss my results on the binary synthetic data and the realistic one respectively, with further analysis to model performance for each dataset. Finally, I summarise my findings in §4.6.

	#Docs	#Synthetic Docs	Avg #Sents	Avg Sent Len
Train	1,238	23,174	21.4	24.1
Dev	138	2,593	18.13	25.13
Test	1,090	20,766	21.9	24.28

Table 4.1: Statistics for the WSJ data. ‘#Docs’ represents the number of original articles and ‘#Synthetic Docs’ the number of original articles + their permuted versions. ‘Avg #Sents’ is the average number of sentences per document.

4.1 Datasets and preprocessing

4.1.1 The Wall Street Journal - synthetic data

Following prior work (Elsner and Charniak, 2008, 2011b; Lin et al., 2011; Tien Nguyen and Joty, 2017), I use the Wall Street Journal (WSJ) portion of Penn Treebank, which contains business-focused articles, for the binary discrimination task. I prefer it to other widely used datasets (e.g., Earthquakes and Accidents (Barzilay and Lapata, 2008)) as it contains longer articles, allowing the models to reason over long stretches of discourse. Additionally, each of the Earthquakes and Accidents corpora contains 100 original articles for training and 100 for testing while the WSJ contains 1,376 for training and 1,090 for testing.¹ I follow the work of Tien Nguyen and Joty (2017) and use sections 00 – 13 of the WSJ for training and 14 – 24 for testing (each section contains 100 documents); I also remove the documents consisting of one sentence. Following previous studies (Barzilay and Lapata, 2008; Elsner and Charniak, 2008), I create 20 permutations per document by randomly shuffling its sentences, making sure to exclude duplicates² or versions that happen to have the same ordering of sentences as the original article. I also follow previous work and do not account for paragraph boundaries in this corpus. The documents are annotated with binary labels, where an original document is considered coherent and given a score of one and each shuffled document is incoherent and assigned a zero score. As for tokenisation, the available version of the WSJ dataset is already tokenised and sentence boundaries are detected, so I leverage that. For training, I follow the same train-dev split of Tien Nguyen and Joty (2017) which is a 9 : 1 split.³ All words are lowercased and, for the standard word embeddings setup, I follow the traditional method of mapping the words that occur once in the training data to a special unknown token $\langle UNK \rangle$ (Collobert et al., 2011). The vocabulary training size (i.e., the number of unique tokens) is 30,048 word. The statistics for the WSJ corpus are revealed in Table 4.1.

In order to evaluate model performance on the WSJ, I again follow previous work (Barzilay and Lapata, 2008) and calculate the pairwise ranking accuracy (PRA) between an

¹The counts are calculated after excluding documents with a single sentence.

²This means that very short documents might have less than 20 permutations.

³https://github.com/datienguyen/cnn_coherence/tree/master/data

	Yahoo			Clinton			Enron		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
#Documents	900	100	200	900	100	200	900	100	200
Avg #Sents	7.5	7.3	7.4	6.6	6.3	6.6	7.6	7.8	7.8
Avg #Paras	1.3	1.3	1.3	4.1	4.3	4.0	2.8	3.0	2.8
Avg Sent Len	20.8	20.5	21.7	27.8	28.7	28.2	24.1	24.4	24.4
Avg Para Len	6.1	5.7	5.7	2.3	2.2	2.3	3.3	3.2	3.3
Training Vocab Size	10,731			12,658			11,169		
QWK	0.386 ± 0.009			0.250 ± 0.011			0.273 ± 0.011		
Class Dist. (%)	45.2, 17.5, 37.2			28.6, 21.2, 50.2			30.0, 19.3, 50.6		

Table 4.2: Statistics for the GCDC datasets. ‘Avg #Sents’ and ‘Avg #Paras’ represent the average number of sentences and paragraphs in a document respectively. ‘Avg Sent Len’ is the average number of words per sentence and ‘Avg Para Len’ is the average number of sentences per paragraph. ‘Training Vocab Size’ represents the number of unique words in the training set. QWK represents the agreement for expert annotators (mean and standard deviation, calculated using leave-one-out resampling). The three numbers in ‘Class Dist.’ denote the percentage of the low, medium and high classes respectively, using the consensus labels of expert annotations.

original text and its 20 permuted counterparts. Additionally, I use the more generalised total pairwise ranking accuracy (TPRA) which supports a more rigorous evaluation by verifying if a model can do more than comparing different distributions of the same set of entities and can generalise better across various documents. TPRA also facilitates assessing model susceptibility to text length (PRA only compares a document to its different versions that are of the same length). For more details about PRA and TPRA, see §2.7.

4.1.2 The Grammarly Corpus of Discourse Coherence - realistic data

The Grammarly Corpus of Discourse Coherence (GCDC)⁴ is a dataset that contains emails and online posts written by non-professional writers with varying degrees of proficiency and care (Lai and Tetreault, 2018). Specifically, the dataset contains texts from four domains: **Yahoo** online forum posts, emails from Hillary **Clinton**’s office, emails from **Enron** and **Yelp** business reviews. As some of the reviews from the latter were removed by Yelp at the time I developed my models, in addition to having a *slight* expert inter-annotator QWK agreement (see Table 4.2), I evaluate my models on each of the first three domains that exhibit *fair* agreement (for more details about the strength of agreement, see Landis and Koch (1977)). Lai and Tetreault (2018) calculated agreement using leave-one-out resampling (§2.7) and averaging the QWK values across 1,000 runs following Pavlick and Tetreault (2016). Expert and untrained Amazon Mechanical Turk raters were asked to

⁴<https://github.com/aylai/GCDC-corpus>

annotate each document with a score $\in \{1, 2, 3\}$, representing low, medium and high levels of coherence respectively. The untrained inter-annotator agreement was quite low, and therefore, I follow Lai and Tetreault (2018) and only use the expert annotations. These annotations were done by three experts with “previous annotation experience” and no solid instructions were provided. Lai and Tetreault (2018) described how they guided the expert raters as follows: “We provided a high-level description of coherence but no detailed rubric, as we wanted them to use their own judgment. We also provided examples of low, medium, and high coherence along with a brief justification for each label.”

For my experiments, I particularly use the consensus rating of the expert scores as calculated by Lai and Tetreault (2018) (by averaging the raters’ scores then thresholding the mean coherence score: low $\leq 1.8 <$ medium $\leq 2.2 <$ high). The models are evaluated using three-way classification accuracy to test their ability to predict the correct coherence class. I use the same train and test sets as Lai and Tetreault (2018) and follow them by dividing the training documents into train-dev splits with a 9 : 1 ratio. I use spaCy (Honnibal and Johnson, 2015) for tokenisation and sentence boundary detection. The dataset is already annotated with paragraph boundaries so I leverage that following Lai and Tetreault (2018), and apply the paragraph equations in §3.1.3. Similar to the WSJ, all the tokens are lowercased and words that occur once in the training data are represented with $\langle UNK \rangle$. Statistics for the three GCDC datasets I use are displayed in Table 4.2, and I show examples from the Yahoo domain in Appendix A.

4.2 Previous neural models

In this section, I present previous neural coherence models that I compare my approach to. Further details about the models are provided in §2.4.1

4.2.1 Local coherence

I compare my approach to the local coherence (LC) model (Li and Hovy, 2014), using an LSTM sentence encoder (Li and Jurafsky, 2017). The LC model is depicted in Fig. 4.1 and its training hyperparameters are detailed in §4.3. First, the model builds sentence embeddings by applying an LSTM (Eq. 2.2) and taking the hidden state of the last word as the sentence vector. A window approach then applies a filter of weights $\in \mathbb{R}^{l \times dim^s \times dim^c}$ over clique embeddings, produced by concatenating vectors of adjacent sentences, to extract clique representations (Eq. 2.5), where l is a hyperparameter indicating the window size, dim^s is the length of the sentence vector and dim^c denotes the convolution output size. Clique representations are scored by a linear operation followed by sigmoid. A clique is assigned a score of 1 if it is coherent (i.e., its sentences are *not* shuffled) and 0 if it is incoherent (i.e., its sentences are shuffled). The network optimises its parameters to

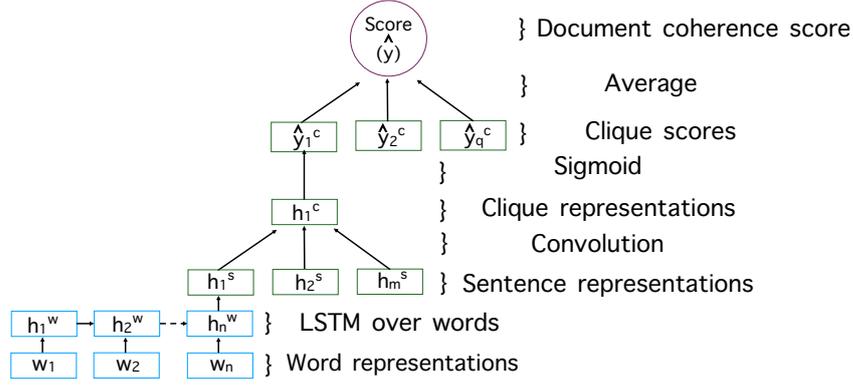


Figure 4.1: The LC model architecture using a window of size 3. All h^s representations are computed the same way as h_1^s . The figure depicts the process of predicting the first clique score, which is applied to all the cliques in the text. The output coherence score is the average of all the clique scores and q is the number of cliques.

minimise the negative log-likelihood of the clique gold scores (y^c), given the network’s predicted scores (\hat{y}^c):

$$L_{lc} = \frac{1}{q} \sum_{k=1}^q [-y_k^c \log(\hat{y}_k^c) - (1 - y_k^c) \log(1 - \hat{y}_k^c)] \quad (4.1)$$

where q is number of cliques in text. The final coherence score of a document is calculated as the average of all of its clique scores (Li and Jurafsky, 2017):

$$\hat{y} = \frac{1}{q} \sum_{k=1}^q \hat{y}_k^c \quad (4.2)$$

This is in contrast to Li and Hovy (2014) who multiplied the estimated clique scores to generate the overall document score. This means that if only one clique is misclassified as incoherent and assigned a score of 0, the whole document is regarded as incoherent. I aim to soften this assumption and use the average instead to facilitate modeling more fine-grained degrees of coherence. I apply ‘*valid*’ convolution (Goodfellow et al., 2016, p. 343) where $q = m - l + 1$, m is the number of sentences and l is the window size. I evaluate the LC model on both the WSJ and the GCDC.⁵ On the latter, the LC model achieved the highest classification accuracy on Clinton and Enron datasets (Lai and Tetreault, 2018).

4.2.2 Neural EGrid

Neural EGrid models constitute a strong approach to compare my MTL model to as they focus on subject-object-other roles for entities while being able to capture long-range

⁵For the WSJ, I implement, train and test the model, whereas for the GCDC, I report the results of Lai and Tetreault (2018).

transitions. This allows us to examine the advantage of predicting all GRs with MTL as well as building sentence representations from all the words in input sentences, not just entities. I utilise two EGrid models: CNN-EGrid_{ext} (Tien Nguyen and Joty, 2017) and CNN-EGrid_{lex} (Joty et al., 2018).

Extended CNN EGrid (CNN-EGrid_{ext}) I leverage the CNN-EGrid_{ext} which applies a CNN followed by max pooling over EGrid text representations that are extended with entity specific features. Training is achieved in a pairwise fashion where the network is given a pair of document grids (a coherent one (g_i) and its incoherent counterpart (g_j)) and optimises the following margin/ranking objective that aims at maximising the margin between coherent and incoherent documents:

$$L_{egrid}(\theta) = \max\{0, 1 - f(g_i|\theta) + f(g_j|\theta)\} \quad (4.3)$$

where θ is the model parameters. I use the public implementation by Tien Nguyen and Joty (2017)⁶ and extract the EGrid representations using the Brown coherence toolkit (Elsner and Charniak, 2011b).⁷

Lexicalised CNN EGrid (CNN-EGrid_{lex}) CNN-EGrid_{lex} uses the same framework as CNN-EGrid_{ext}, yet the core difference is that it integrates lexical information about entities by representing each entity with its word embedding (specifically, Google pre-trained embeddings (Mikolov et al., 2013c)) together with its subject-object-other role. CNN-EGrid_{lex} also excludes the three entity-specific features used by CNN-EGrid_{ext}. I use the public code by Joty et al. (2018).⁸

Both EGrid models are well-suited for binary data as they are trained in a pairwise fashion, where the input for the model consists of a pair of documents (a coherent document and its incoherent counterpart). However, the available implementation of the models needs to be modified to accommodate for the multiple classes in the realistic data, which would be an interesting avenue for future work. In this thesis, I only utilise them in the binary synthetic domain.

4.2.3 Local coherence discriminator

I use the local coherence discriminator (LCD) approach (Xu et al., 2019) which encodes sentences and adds a discriminative layer over each sentence pair to distinguish between coherent and incoherent pairs. The network also optimises a margin loss (Eq. 4.3), given coherent and incoherent pairs of sentences. Since the training strategy of the LCD approach

⁶https://github.com/datienguyen/cnn_coherence/

⁷<https://bitbucket.org/melsner/browncoherence>

⁸<https://ntunlpsg.github.io/project/coherence/n-coh-acl18/>

is pairwise, similar to the neural EGrid models, I only evaluate it on the WSJ. I use three sentence encoders, as will be described next, and utilise the public implementation of Xu et al. (2019).⁹

LCD with language modeling This model (LCD-L) uses an RNN language model encoder. It is, overall, the best model proposed by Xu et al. (2019) and it achieves the published state-of-the-art results on the WSJ using the PRA metric.

LCD with fastText I create a variant of the LCD model using fastText embeddings (Mikolov et al., 2018) (§2.3.2) that were proven to be efficient in various NLP tasks (Joulin et al., 2016), even when used as bag-of-words baselines (Conneau et al., 2018). The LCD-fastText model encodes each sentence by simply averaging the fastText vectors of its words.

LCD with ELMo This version of the model builds sentence representations by averaging the ELMo vectors of their words (LCD-ELMo). ELMo embeddings are created by taking the last layer for each word representation (§3.1.1)

LCD with BERT Similarly, I encode sentences by averaging their BERT embeddings (LCD-BERT), created as described in §3.1.1.

4.2.4 Paragraph sequence

Lai and Tetreault (2018) implemented the paragraph sequence (PARSEQ) model, which is a hierarchical neural network consisting of three LSTMs to generate sentence, paragraph and document representations. The network architecture is similar to my STL model; the key difference is that I use a Bi-LSTM and aggregate the representations produced at different network levels with attention. I compare my models to PARSEQ on the GCDC as it achieved state-of-the-art results on Yahoo dataset.

4.3 Training and hyperparameters

I implement and train the models described in Sections 3.1, 3.2, 3.3 and 4.2.1, using Keras v. 2.2.4 (Chollet et al., 2015); I use its default initialisation settings for network parameters. All the non-contextualised models are initialised with GloVe embeddings and words that do not exist in the pre-trained embedding space are initialised randomly with values drawn from a normal distribution with mean = 0 and scale = 0.01. For regularisation, I apply dropout (Hinton et al., 2012) with probability 0.5, which is a value typically used in

⁹https://github.com/BorealisAI/cross_domain_coherence

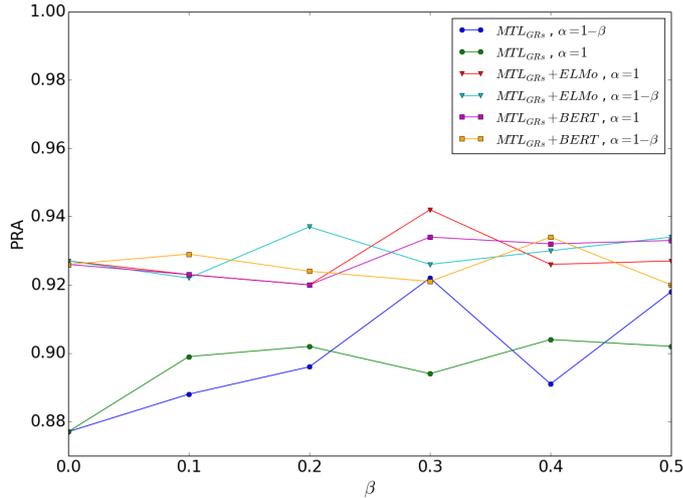


Figure 4.2: PRA value (shown on the y-axis) for MTL_{GRs}, MTL_{GRs}+ELMo and MTL_{GRs}+BERT models with different α and β values, where β is shown on the x-axis and α is shown in the legend. I report the results when α is fixed at 1 and β changes and when $\alpha + \beta = 1$.

literature (Baldi and Sadowski, 2013). Dropout is applied at two layers in any model; the first is the word embeddings layer, and the second is the output of the convolutional operation for the LC model or the output of the sentence Bi-LSTM for all the other models (Eq. 3.1). For optimisation, I use RMSProp (Tieleman and Hinton, 2012) with the default Keras values: learning rate = 0.001 and *gradient moving average decay factor* (ρ) = 0.9. I sort the input documents by length (i.e., the number of sentences in the document), and create mini-batches of size 32. In order to unify the lengths of the sentences and documents in the same mini-batch, I first calculate the maximum document length (d_{max}), as well as the maximum sentence length (s_{max}), in each mini-batch. I then pad any sentence of length $< s_{max}$ with zero vectors of the dimensionality of the word embeddings (d^w), and pad any document of length $< d_{max}$ with zero vectors $\in \mathbb{R}^{s_{max} \times d^w}$. The same padding technique is applied for paragraphs in the GCDC.

All the WSJ models are trained for 30 epochs and the GCDC ones for 20 epochs, as they converge early. Performance is monitored on the development sets and, for the final evaluation, I select the model that yields the highest PRA for the WSJ and highest classification accuracy for the GCDC. I search for the optimal size of the hidden layers in the space $\{100, 200, 300\}$ and GloVe embeddings in $\{50, 100, 300\}$; tuning is done for each dataset separately for the STL model, as the base model, then applied to the different models that inherit from it.

As for the loss weights in MTL, since there are many possible combination of values, I limit my tuning to two settings:

- I fix the main loss weight (α) at 1 and tune the auxiliary loss weight (β) using the

	d^w		h^w	h^s	h^p	α		β
	GloVe	ELMo/BERT				MTL	MTL+ELMo/MTL+BERT	
WSJ	50	1,024	100	100	-	0.7	1	0.3
Yahoo	300	1,024	100	100	100	1	1	0.1
Clinton	300	1,024	100	200	100	1	1	0.1
Enron	300	1,024	100	100	100	1	1	0.2

Table 4.3: Model hyperparameters: d^w denotes the dimensionality of word embeddings (1,024 is the default value for ELMo and BERT_{LARGE}); the h hyperparameters refer to the size of LSTM hidden layers with the superscripts w , s and p referring to word, sentence and paragraph hidden layers respectively; α is the main loss weight and β the secondary one. The values in the table are the final hyperparameters (where applicable) for all STL and MTL models, with the exception of MTL_{GRs+POS} where $\alpha = 0.8$, $\beta_1 = 0.1$ and $\beta_2 = 0.1$. MTL values are for both the GR and POS models.

values $\{0.1, 0.2, 0.3, 0.4, 0.5\}$.

- I interpolate the α and β values in $[0, 1]$ with a step of 0.1 and the constraint $\alpha + \beta = 1$. This constraint is modified for the model that utilises two secondary losses (MTL_{GRs+POS}) to be: $\alpha + \beta_1 + \beta_2 = 1$

In Fig. 4.2, I depict the change in performance (PRA) of the GR-based models (MTL_{GRs}, MTL_{GRs+ELMo} and MTL_{GRs+BERT}) on the WSJ dev set when the loss weights change. Regarding the MTL_{GRs+POS} model, the final weight values I use are: $\alpha = 0.8$, $\beta_1 = 0.1$ and $\beta_2 = 0.1$. Table 4.3 summarises the best STL and MTL hyperparameter values used for input and hidden layer dimensionality as well as weight losses.

As for the LC model, on the WSJ, I follow Li and Hovy (2014) and Li and Jurafsky (2017) and set the size of the hidden layer to 100 and the window size to 3, while on the GCDC I report the model results from Lai and Tetreault (2018) as I use their same test set. Similarly, for the PARSEQ model on the GCDC, I report the results from Lai and Tetreault (2018). Regarding the LCD and EGrid models, as mentioned earlier, I use their available public implementations and the default hyperparameters described in their respective papers. As for the concatenation models (Concat_{GRs} and Concat_{POS}), the embedding size for the syntactic feature is set to 50.

In order to reduce model variance, for all the models I run the WSJ experiments 5 times with different random initialisations and the GCDC ones 10 times (following Lai and Tetreault (2018)), and average the predicted scores of the ensembles for the final evaluation. Averaging ensemble predictions has been widely leveraged in deep learning approaches and shown to reduce error rates committed by individual neural networks (Krizhevsky et al., 2012; Sutskever et al., 2014; Taghipour and Ng, 2016).

Annotating the input words with their GRs, for the GR-based models, is achieved using the Stanford Dependency Parser (v. 3.8) (Chen and Manning, 2014), where the parser extracts 39 different GRs (of UD and their subtypes). All the GR-based models

leverage the full set of extracted GRs, except for the MTL_{SOX} model that only uses subject and object roles. As for the POS-based models, I parse the input with the Stanford POS tagger (v. 3.8) (Toutanova et al., 2003) that leverages the Penn Treebank tagset and extract a total of 41 different tags. The full list of the extracted GRs and POS tags is shown in Table 3.1.

4.4 Binary experiments

In this section, I present the baselines that I include in my binary experiments on the WSJ and discuss the results of the different models. I also conduct a further analysis to explicate the obtained results.

4.4.1 Baselines

For all the baselines introduced in this section, I use the hyperparameters and training setup described in §4.3.

MTL with random syntactic labels The MTL framework optimises a main loss function and an auxiliary one and assigns different weights to each function based on its importance for the final prediction. This combination of losses might raise the question of whether the model actually benefits from predicting syntactic labels, or the addition of the auxiliary function works as a regularisation mechanism (i.e., a penalty on the main loss) that boosts the performance. In order to further examine this question, I create a version of MTL_{GRs} and MTL_{POS} where I randomly shuffle the syntactic labels for each sentence, thus each word will be mapped to an incorrect label. If the performance does not get affected this would show that the auxiliary loss might just be regularising the network and there is no value for learning the GR or POS labels; otherwise, the usefulness of learning syntactic information could be validated. I refer to the models with randomised GRs and POS tags as $MTL_{GRs-rand}$ and $MTL_{POS-rand}$ respectively.

STL with untuned embeddings In the models that use standard embeddings, all the parameters are tuned with back-propagation, where the error gradients are back-propagated to the word embedding layer. Fine-tuning word representations is useful as it modifies the input embedding space to be more tailored for the task. In order to examine the merit of this, I create another setup of the STL model, trained on the WSJ, where I keep the pre-trained GloVe representations fixed during training (I refer to it as *STL (untuned embed)*), and compare it to the setup where I fine-tune the embeddings. Furthermore, to better understand the effect of fine-tuning, I select a few finance-related terms from the WSJ vocabulary and report, in Table 4.4, the most similar word to each term in both the

Word	Fine-tuned	Untuned	Freq
capital	money	central	6,268
bank	agency	banks	12,145
banks	market	bank	5,112
drop	losses	dropping	2,842
fall	directors	rise	2,354
store	containers	shop	1,894
properties	third-quarter	estate	1,134
bonds	agreement	bond	10,107
stocks	dividend	stock	8,070

Table 4.4: The table displays the most similar word in the fine-tuned and untuned embedding space for the word in the left. The untuned space is fixed GloVe embeddings of size 50, whereas the fine-tuned space is the result of initialising the STL model with the GloVe embeddings and fine-tuning them by training the model on the WSJ. ‘Freq’ is the frequency of the word in the training set.

fine-tuned and untuned spaces. Similarity between two words w_1 and w_2 is defined as the cosine similarity between the two vectors representing these words as follows:

$$sim(w_1, w_2) = \cos(\theta) = \frac{\vec{w}_1 \cdot \vec{w}_2}{\|\vec{w}_1\| \cdot \|\vec{w}_2\|} \quad (4.4)$$

where θ is the angle between the two vectors. In order to perform this analysis, I use *Gensim* Python scripts (Řehůřek and Sojka, 2010).

STL with averaged ELMo embeddings As explained in §3.1.1, I integrate ELMo embeddings either by using the top layer in the three-layer representation or averaging the three layers. I use the two techniques with the STL model on the WSJ and carry on, in the rest of the experiments, with the top-layer approach as it achieves the higher performance.

4.4.2 Results

The results for the binary discrimination task on the WSJ corpus are shown in Table 4.5. Significance is calculated using a randomisation test (Yeh, 2000), with p-value < 0.01.¹⁰ We further explain the obtained PRA and TPRA results as follows.

¹⁰A randomisation test calculates the difference in accuracy (PRA or TPRA) of two systems, A and B, if their predictions are randomly exchanged with a probability of 0.5. This process is repeated for R iterations (we set $R = 1,000$) and in each iteration, if the absolute value of the difference between the system accuracies after randomisation is greater than or equal the absolute value of the difference between the system accuracies before randomisation, we increment a variable c by 1. The p-value is then calculated as: $p = \frac{c+1}{R+1}$.

Model	PRA	TPRA	r
Egrid CNN _{ext}	0.876	0.656	0.033
Egrid CNN _{lex}	0.846	0.566	-0.030
LC	0.741	0.728	0.075
STL	0.877	0.893	0.225
STL (untuned embed)	0.768	0.781	0.069
Concat _{GRs}	0.896	0.908	0.226
Concat _{POS}	0.895	0.904	0.222
MTL _{GRs-rand}	0.911	0.920	0.195
MTL _{POS-rand}	0.919	0.928	0.167
MTL _{GRs}	0.932	0.941	0.260
MTL _{POS}	0.934	0.942	0.223
MTL _{SOX}	0.899	0.913	0.231
MTL _{GRs+POS}	0.930	0.937	0.164
STL+ELMo	0.953	0.965	0.227
STL+ELMo-avg	0.948	0.960	0.217
MTL _{GRs} +ELMo	0.960	0.969*	0.234
MTL _{POS} +ELMo	0.959	0.969*	0.227
STL+BERT	0.954	0.961	0.271
MTL _{GRs} +BERT	0.961	0.968	0.239
MTL _{POS} +BERT	0.960	0.969*	0.257
LCD-L	0.945	0.870	0.313
LCD-fastText	0.940	0.843	0.184
LCD-ELMo	0.968	0.931	0.319
LCD-BERT	0.971*	0.922	0.339

Table 4.5: The middle column shows the results of the binary discrimination task on the WSJ. * indicates significance (p-value < 0.01) over all the other models, except LCD-ELMo for PRA and MTL_{GRs}+BERT for TPRA, based on the randomisation test. The last column shows the Pearson’s correlation r between the similarity of incoherent documents to their original counterpart (calculated as will be discussed in §4.4.3.1), and the coherence scores assigned to these incoherent documents.

Using contextualised embeddings From the results, we can see that adding contextualised embeddings always significantly improves the performance over standard vectors; the top PRA is 0.971 obtained by LCD-BERT and the top TPRA is 0.969 yielded by MTL_{GRs}+ELMo, MTL_{POS}+ELMo and MTL_{POS}+BERT. This further motivates the value of using contextualised representations in downstream tasks due to their ability to capture syntactic and semantic information. As for the comparison between ELMo and BERT, we find that both embeddings perform closely when utilised in STL or MTL setups. The exception is the significant difference between LCD-ELMo and LCD-BERT on the TPRA metric (0.931 and 0.922 respectively). This could be explained by the fact that these LCD models optimise a smaller number of parameters (only the final MLP layer) in comparison to the other models, which makes the impact of the different contextualised models more

notable. As for the different ELMo initialisation techniques, we can see that using the last layer of ELMo (STL+ELMo) surpasses the averaging method (STL+ELMo-avg) and thus I employ the former in all the ELMo-based approaches.

Using MTL We observe that incorporating syntactic information in an MTL fashion, significantly boosts the performance of the models over their STL counterparts.¹¹ This also applies to the ELMo and BERT versions of MTL_{GRs} and MTL_{POS} versus their STL versions (STL+ELMo and STL+BERT), which suggests that despite the syntactic information captured by contextualised embeddings, the models can still benefit from learning syntactic properties as a secondary task. However, the impact of MTL becomes smaller when ELMo or BERT are leveraged (e.g., PRA of STL vs. MTL_{GRs} is 0.877 vs. 0.932 while STL+ELMo vs. MTL_{GRs}+ELMo is 0.953 vs. 0.960, and STL+BERT vs. MTL_{GRs}+BERT is 0.954 vs. 0.961), further corroborating the value of contextualised embeddings in capturing syntactic features.

Additionally, Table 4.5 reveals the superiority of MTL_{GRs} and MTL_{POS} over Concat_{GRs} and Concat_{POS}. This demonstrates that learning syntactic labels within an MTL framework facilitates building linguistically richer representations, in comparison to forcing these labels as inputs. In addition, MTL limits syntactic parsing to training time. As for the weights assigned to the main and auxiliary tasks in MTL, we can see in Fig. 4.2 that the relation between α and β on one hand and performance (PRA) on the other is non-monotonic and the figure does not exhibit a consistent pattern; hence, it is important to tune these weights to find balance between the tasks.

Moreover, we find that MTL_{GRs} and MTL_{POS} perform significantly better than their randomised versions (MTL_{GRs}-rand and MTL_{POS}-rand), further validating the MTL approach and demonstrating that useful features could be learned from grammatical and POS labels. Nonetheless, the randomised models still yield a high performance (PRA for MTL_{GRs}-rand = 0.911 and for MTL_{POS}-rand = 0.919) and thus I recommend them as strong baselines for MTL frameworks in NLP tasks.

In terms of the MTL GR-based models and their POS-based counterparts, we interestingly find that they provide very close performance (even identical in some cases). Furthermore, combining the two types of syntactic features in MTL_{GRs+POS} doesn't yield any performance gains. This raises the question of whether the models learn complementary features when leveraging different syntactic properties which I further investigate in §4.4.3.3. We also observe that utilising the full set of GRs (MTL_{GRs}) significantly outperforms focusing on subject and object roles (MTL_{SOX}) which I further discuss in §4.4.3.2.

¹¹Statistical significance is calculated using randomisation test with p-value < 0.01.

Fine-tuning standard word embeddings The value of fine-tuning the word embeddings is indicated by the substantial difference in performance between STL and its untuned version. Table 4.4 gives a few examples that illustrate how the embedding space gets shifted by fine-tuning. For example, the closest word to ‘*capital*’ is ‘*money*’ in the fine-tuned space and ‘*central*’ in the untuned one, showing how the word gets closer to its financial sense by fine-tuning. However, in other cases, fine-tuning pushes some words towards semantically less related terms such as detecting ‘*directors*’ as the most similar word to ‘*fall*’, in contrast to ‘*rise*’ in the untuned space.

Other neural models Table 4.5 also reveals that neural EGrid approaches underperform other hierarchical and LCD-based models. Specifically, they do not generalise when documents are compared against counterparts from the whole test set (TPRA for Egrid $\text{CNN}_{\text{ext}} = 0.656$ and for Egrid $\text{CNN}_{\text{lex}} = 0.566$). This could be partly attributed to the pairwise training strategy adopted by these models and their inability to compare entity-transition patterns across different topics.

As for the LCD-based methods, the results demonstrate the efficacy of the approach with different sentence encoders, particularly with contextualised embeddings. The highest PRA value across all the models is 0.971 obtained by LCD-BERT, significantly surpassing Xu et al.’s 2019 published state-of-the-art LCD-L approach. However, my MTL approach (with or without contextualised embeddings) generalises better to TPRA metric. In general, the PRA obtained by the LCD-based models suggest the effectiveness of the approach, even LCD-fastText that simply averages fastText vectors. This family of models captures local coherence by utilising a number of linear operations (concatenation, element-wise difference, element-wise product and absolute value of element-wise difference) which increases their expressive power and facilitates the learning of richer representations. Furthermore, creating the incoherent examples by negative sampling enables the LCD-based models to effectively learn from a large space of negative examples (§2.4.1). On the other hand, the LC model that also captures local coherence, but using an LSTM sentence encoder in a CNN network, significantly underperforms all LCD models. This shows that focusing on local coherence can be effective based on the model architecture. In addition, the comparatively low performance by LC could be attributed to the simplicity of the approach: LC utilises no attention mechanism as the MTL and STL family of models do, nor has expressive enough transformations or uses a sampling strategy as LCD models do. Finally, the PRA of LCD-L (0.945) further attests to the conclusions by Xu et al. (2019) that discriminative and generative approaches could be successfully utilised together for coherence modeling; however, its relatively low TPRA (0.870) indicates the limitation of the approach in comparing documents of different topics.

4.4.3 Analysis

In this section, I further analyse the results above to have a better understanding of the models.

4.4.3.1 Sensitivity to sentence order

Creating incoherent documents by distorting the sentence order in the original ones, would result in incoherent texts; however, the permuted versions would vary in their degree of incoherence. For instance, a random permutation might result in only one sentence being out of place while another permutation might completely mess up the sentence order. A robust coherence scoring system should be able to distinguish between the two cases and capture variant levels of coherence. In order to assess that, I measure the correlation between the scores predicted by the different models for incoherent documents and the degree of similarity between these documents and their original version. I use Kendall’s τ to estimate the degree of similarity between a permuted document and its original counterpart (Lapata, 2006), as explained in §2.7. Lapata (2006) have also shown that Kendall’s τ ranks correlate with human judgement of the overall text understandability and coherence, which further motivates its usage to evaluate the sensitivity of models to different ranks of (in)coherence. After calculating the Kendall’s τ scores (i.e., the scores that indicate similarity to the original document), I calculate the linear relationship between these scores and the coherence scores generated by the models by measuring Pearson’s correlation coefficient (r) between the two variables;¹² I report the results in Table 4.5. The results show that most of the models have positive correlation with the degree of similarity with the original text, demonstrating some potential to capture different coherence levels. However, the exception to that are the EGrid models, STL (untuned embed) and the LC, which agrees with their comparatively low performance in the binary task (shown by their PRA and TPRA values). In other words, the models that perform the poorest on the binary discrimination task achieve the lowest Pearson’s correlation with the degree of similarity measure, which is indicative of their inability to model partial coherence. The strongest correlation is yielded by LCD-BERT which achieves the highest PRA as well. Despite the competitive performance of LCD-fastText and MTL_{GRs+POS}, there is a drop in their correlations in comparison to their family of models (LCD and MTL respectively); however, it is not clear to us why this drop happens.

In order to further test the ability of models to capture more fine-grained coherence ranks, it would be interesting in the future to train the models on multiple ranks that denote their similarity to the original document (Feng and Hirst, 2012) or partially permuted

¹²Pearson’s r is the same measurement used by Lapata (2006) to estimate the relationship between the Kendall’s τ scores and the human ratings that measure the (in)coherence of documents.

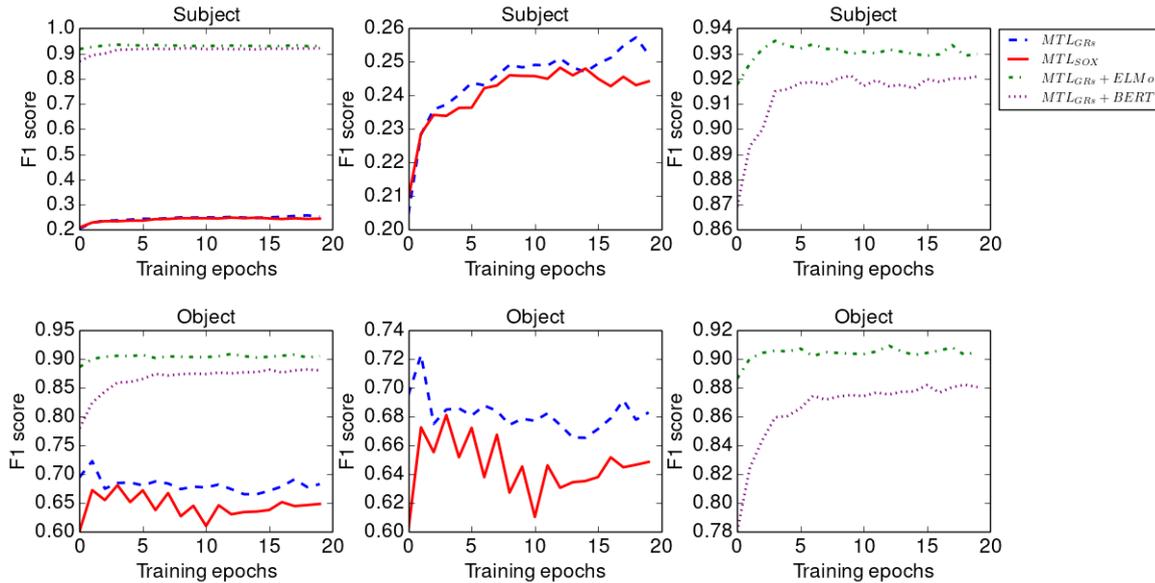


Figure 4.3: F1 scores (on the y-axis) for subject and object predictions with the GR-based models (MTL_{GRs} , MTL_{SOX} , $MTL_{GRs}+ELMo$ and $MTL_{GRs}+BERT$) over the first 20 epochs of training (on the x-axis). The two graphs on the left depicts the scores of the four models, the two in the middle are a close-up on the MTL_{GRs} and MTL_{SOX} scores, and the two on the right are a close-up on the $MTL_{GRs}+ELMo$ and $MTL_{GRs}+BERT$ scores. The graphs are based on the WSJ dev set.

documents (Moon et al., 2019).

4.4.3.2 Subject and object prediction

Table 4.5 shows that leveraging all GR types for MTL surpasses using a subset that only focuses on salient roles (MTL_{SOX}). Additionally, enhancing GR prediction with ELMo or BERT vectors further improves the performance on the main task of coherence evaluation. I further examine these results by investigating the performance of these models on the secondary task of GR prediction and plot in Fig. 4.3 the F1 scores over the training epochs for predicting the subject and object types using the GR-based approaches: MTL_{GRs} , MTL_{SOX} , $MTL_{GRs}+ELMo$ and $MTL_{GRs}+BERT$. I particularly analyse subject and object roles as they are strong indicators of entity saliency (Grosz et al., 1995; Kameyama, 1998; Barzilay and Lapata, 2008). From the figure, we find that the difference between the F1 scores (especially in object prediction) obtained by MTL_{GRs} and MTL_{SOX} (which could be better seen in the two middle graphs in Fig. 4.3) indicates that learning to predict a larger set of GR types boosts the predictive power of the model for the subject and object types, corroborating the value of entity-based properties for coherence. Furthermore, it’s obvious from the two leftmost graphs that using contextualised embeddings substantially improves the model’s ability to identify subject and object roles due to the rich syntactic

information carried by these embeddings. Finally, the rightmost graphs show that ELMo has a better predictive power than BERT. I conjecture, however, that the layers leveraged from ELMo (last layer) and BERT (layer 16) play an important role in determining their ability to capture word-level features, and using different layers and interpolations might lead to different results.

4.4.3.3 Attention visualisation

I further investigate the features learned by my models by analysing the attention weights they calculate (a_{it}^w in Eq. 3.2). We can think of each model as a reader who processes a text sentence by sentence and focuses on certain parts of the discourse, as previously discussed in §2.1.4. Examining the attention weights would help us understand what the models focus on and the features that contribute the most to their final decision (§2.6). Additionally, it will allow us to compare the various network setups and examine whether relying on GRs, POS tags, ELMo/BERT vectors or just GloVe features impacts the attention of the models, which is particularly helpful in the cases where models give very similar results such as MTL_{GRs} and MTL_{POS} . I conduct two types of analysis: quantitative and qualitative, and the models I analyse are STL, MTL_{GRs} , MTL_{POS} and their ELMo and BERT versions. My analysis is performed on the coherent documents in the dev set of the WSJ.

Quantitative analysis. For this analysis, I examine whether a model gives more attention to specific syntactic labels (GRs or POS tags), where the labels I analyse are the ones extracted by the parser. To that end, I calculate two scores for each label: an *importance* score (I) and a *rank* one (R).

The **Importance** (I) score for a label indicates its impact in determining the final coherence score in terms of the attention weights given to this label. For example, if a label is assigned $I = 0.25$, this means that the network gives 25% of its attention to this label and hence it highly influences the final decision. I calculate the I score of label l for each document d (I_d^l) in the dataset as follows:

$$I_d^l = \frac{1}{N} \sum_t a_{ild}^w \quad (4.5)$$

where a_{ild}^w is the attention weight for the t -th word, in document d , annotated with the target label l , and N is the number of sentences in d which also denotes the sum of attention weights in d .¹³ In order to calculate the final I score for each label, I average

¹³I note that the sum of attention weights in d equals the number of sentences in d since the sum of attention weights in each sentence is 1.

the scores it obtains across the documents:

$$I^l = \frac{1}{D} \sum_d I_d^l \quad (4.6)$$

where D is the number of documents in the dataset. I calculate the I scores for all the labels for each model and for the purpose of plotting and visualising the scores, I select the 3 highest scoring labels for each model and plot the union set of these labels. I depict two graphs for GRs and POS tags in Fig. 4.4 (a) and (c) respectively.

The **Rank** (R) score refers to the average number of times a label obtains the highest attention weight in a sentence. The R score for label l (R_l) is simply calculated by:

$$R_l = \frac{\text{number of times } l \text{ gets the highest attention weight in a sentence}}{\text{number of sentences in the dataset}} \quad (4.7)$$

I plot the R scores for GRs and POS tags in Fig. 4.4 (b) and (d) respectively.

I first analyse the attention given by the models to GR labels as displayed in Fig. 4.4 (a) and (b). We notice from the figures that the GR-based models (MTL_{GRs}, MTL_{GRs}+ELMo and MTL_{GRs}+BERT) assign the highest attention to words that appear as nominal subjects (*nsubj*). Furthermore, we find that with MTL_{GRs} and MTL_{GRs}+ELMo, the *compound* role comes second using I and R scores. I examine the WSJ training data and find that around 24% of the appearances of the *compound* role (e.g., ‘*asbestos*’ in ‘The asbestos fiber’) are followed by, and dependent on, a word labeled as *nsubj*, which means that these GR-based approaches tend to focus on the subject and/or the nouns modifying it. MTL_{POS} also assigns the highest scores to the *nsubj* role but with lower values than the models that utilise GRs. On the other hand, the ELMo models MTL_{POS}+ELMo and STL+ELMo give the highest importance to *nsubj* and *root* labels,¹⁴ but what is striking is the rank they assign to the *root* role, which stands out clearly in Fig. 4.4 (b). As for MTL_{POS}+BERT, we can see that there is also focus on *nsubj* (it comes in second place); however, the highest scoring label in Fig. 4.4 (a) and (b) is *punct* label for punctuation marks. This could also be seen in the POS tag weights in Fig. 4.4 (c) and (d), where MTL_{POS}+BERT gives attention to the ‘,’ tag higher than any other model. While punctuation marks play an important role in discourse coherence (Dale, 1991a,b), I conjecture that their role is, however, undermined in a synthetic corpus of well-written sentences such as the WSJ (i.e., punctuation marks are not discriminative factors between original and permuted documents). Therefore, it is not clear why MTL_{POS}+BERT attends to punctuation marks. Finally, the STL and STL+BERT approaches seem to be the most distracted ones, giving similar scores to different labels and giving attention to labels that contribute the least to discourse coherence such as *case* labels (e.g., ‘*of*’ in ‘A form of

¹⁴*Root* is usually the main verb.

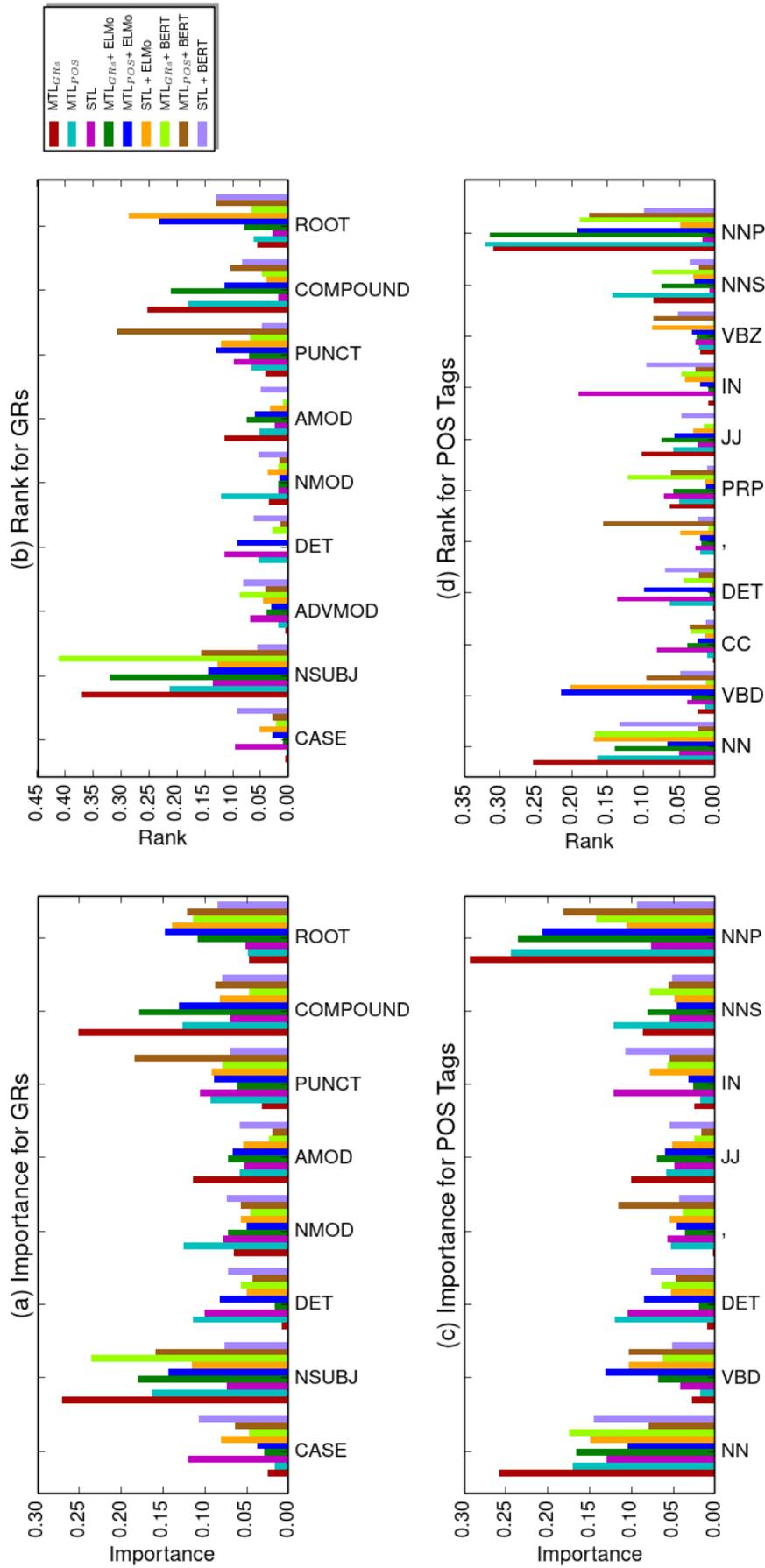


Figure 4.4: Attention analysis on the WSJ dev set. GRs or POS tags are displayed on the x-axis and their corresponding importance and rank scores on the y-axis.

asbestos’). While this behavior is intuitive in STL as it does not incorporate syntactic information, it is less expected in STL+BERT as it relies on BERT representations. In general, the GR attention analysis demonstrates a trend to focus on the subject role, particularly with the models that optimise a GR-based auxiliary loss. These results conform with Centering theory that ranks subject entities as the most salient entities in discourse.

I now investigate how the models attend to the different POS tags as depicted in Fig. 4.4 (c) and (d). The most striking result is the high importance and rank scores given to the proper noun tag (*NNP*) by all the MTL models. This complies with the previously discussed GR attention results as I find that, in the WSJ training data, around 39% of the words labeled as *nsubj* or *compound* are also tagged as *NNP*. Additionally, the nature of the WSJ also justifies the results since the articles are usually about real world entities (e.g., persons or firms) that are realised with proper nouns and hence tagged as *NNP*. The singular noun (*NN*) tag also receives high attention, which is compatible with Centering theory and the attentional state of discourse.¹⁵ We also find that $\text{MTL}_{\text{POS}}+\text{ELMo}$ and $\text{STL}+\text{ELMo}$ give high attention (particularly rank scores) to the *VBD* tag (verb in past tense) similar to their behaviour with GRs where they focus on the *root* of the sentence.

Qualitative analysis. I provide a qualitative analysis for what the models focus on by visualising a few examples from the WSJ dev set. Each example I select consists of two consecutive sentences from a coherent document, and visualisation is done by colour-coding the words in each example based on the attention weight assigned to them.¹⁶ I depict the visualisation in Fig. 4.5 which includes the models discussed in the previous quantitative analysis. If we think of the models as readers of text, as previously mentioned, the highlighted words would constitute the salient parts that help make inferences while processing the text. Each reader/model processes the text differently and thus builds a different conceptual representation of it where the active parts vary from a model to another, as we can see in Fig. 4.5. I interpret the qualitative results in the light of Centering theory, yet I note that, in general, assessing the ability of the models to focus on salient discourse parts could be subjective since there is no one rigid definition of saliency.

Figure 4.5 shows that the qualitative examples further support the quantitative results. I will first start by discussing the MTL models (MTL_{GRs} , and MTL_{POS} and their ELMo and BERT versions) then move to the STL ones (STL , $\text{STL}+\text{ELMo}$ and $\text{STL}+\text{BERT}$). Regarding the MTL models, we can see that they focus on subject words and their dependent compounds. For instance, in example (a), the subject in the first sentence is

¹⁵The plural noun (*NNS*) tag is probably given less attention than the singular one as the latter occurs more than twice as much as the first in the training data.

¹⁶The weights here are a_{it}^w , in Eq. 3.2, assigned to the t -th word in the i -th sentence, not the accumulative I and R scores described in the quantitative analysis.

Example (a)

MTL _{GRs}	Ralston Purina Co. reported a 47 % decline in fourth-quarter earnings , reflecting restructuring costs as well as a more difficult pet food market . The St . Louis company earned \$ 45.2 million , or 65 cents a share , compared with \$ 84.9 million , or \$ 1.24 a share , a year earlier .
MTL _{POS}	Ralston Purina Co. reported a 47 % decline in fourth-quarter earnings , reflecting restructuring costs as well as a more difficult pet food market . The St . Louis company earned \$ 45.2 million , or 65 cents a share , compared with \$ 84.9 million , or \$ 1.24 a share , a year earlier .
STL	Ralston Purina Co. reported a 47 % decline in fourth-quarter earnings , reflecting restructuring costs as well as a more difficult pet food market . The St . Louis company earned \$ 45.2 million , or 65 cents a share , compared with \$ 84.9 million , or \$ 1.24 a share , a year earlier .
MTL _{GRs} + ELMo	Ralston Purina Co. reported a 47 % decline in fourth-quarter earnings , reflecting restructuring costs as well as a more difficult pet food market . The St . Louis company earned \$ 45.2 million , or 65 cents a share , compared with \$ 84.9 million , or \$ 1.24 a share , a year earlier .
MTL _{POS} + ELMo	Ralston Purina Co. reported a 47 % decline in fourth-quarter earnings , reflecting restructuring costs as well as a more difficult pet food market . The St . Louis company earned \$ 45.2 million , or 65 cents a share , compared with \$ 84.9 million , or \$ 1.24 a share , a year earlier .
STL + ELMo	Ralston Purina Co. reported a 47 % decline in fourth-quarter earnings , reflecting restructuring costs as well as a more difficult pet food market . The St . Louis company earned \$ 45.2 million , or 65 cents a share , compared with \$ 84.9 million , or \$ 1.24 a share , a year earlier .
MTL _{GRs} + BERT	Ralston Purina Co. reported a 47 % decline in fourth-quarter earnings , reflecting restructuring costs as well as a more difficult pet food market . The St . Louis company earned \$ 45.2 million , or 65 cents a share , compared with \$ 84.9 million , or \$ 1.24 a share , a year earlier .
MTL _{POS} + BERT	Ralston Purina Co. reported a 47 % decline in fourth-quarter earnings , reflecting restructuring costs as well as a more difficult pet food market . The St . Louis company earned \$ 45.2 million , or 65 cents a share , compared with \$ 84.9 million , or \$ 1.24 a share , a year earlier .
STL + BERT	Ralston Purina Co. reported a 47 % decline in fourth-quarter earnings , reflecting restructuring costs as well as a more difficult pet food market . The St . Louis company earned \$ 45.2 million , or 65 cents a share , compared with \$ 84.9 million , or \$ 1.24 a share , a year earlier .

Example (b)

MTL _{GRs}	Boeing Co. said Trans European Airways ordered a dozen 737 jetliners valued at a total of about \$ 450 million . The 300 and 400 series aircraft will be powered by engines jointly produced by General Electric Co. and Snecma of France .
MTL _{POS}	Boeing Co. said Trans European Airways ordered a dozen 737 jetliners valued at a total of about \$ 450 million . The 300 and 400 series aircraft will be powered by engines jointly produced by General Electric Co. and Snecma of France .
STL	Boeing Co. said Trans European Airways ordered a dozen 737 jetliners valued at a total of about \$ 450 million . The 300 and 400 series aircraft will be powered by engines jointly produced by General Electric Co. and Snecma of France .
MTL _{GRs} + ELMo	Boeing Co. said Trans European Airways ordered a dozen 737 jetliners valued at a total of about \$ 450 million . The 300 and 400 series aircraft will be powered by engines jointly produced by General Electric Co. and Snecma of France .
MTL _{POS} + ELMo	Boeing Co. said Trans European Airways ordered a dozen 737 jetliners valued at a total of about \$ 450 million . The 300 and 400 series aircraft will be powered by engines jointly produced by General Electric Co. and Snecma of France .
STL + ELMo	Boeing Co. said Trans European Airways ordered a dozen 737 jetliners valued at a total of about \$ 450 million . The 300 and 400 series aircraft will be powered by engines jointly produced by General Electric Co. and Snecma of France .
MTL _{GRs} + BERT	Boeing Co. said Trans European Airways ordered a dozen 737 jetliners valued at a total of about \$ 450 million . The 300 and 400 series aircraft will be powered by engines jointly produced by General Electric Co. and Snecma of France .
MTL _{POS} + BERT	Boeing Co. said Trans European Airways ordered a dozen 737 jetliners valued at a total of about \$ 450 million . The 300 and 400 series aircraft will be powered by engines jointly produced by General Electric Co. and Snecma of France .
STL + BERT	Boeing Co. said Trans European Airways ordered a dozen 737 jetliners valued at a total of about \$ 450 million . The 300 and 400 series aircraft will be powered by engines jointly produced by General Electric Co. and Snecma of France .

Example (c)

MTL _{GRs}	The NBC network canceled its first new series of the fall TV season , killing Mel Brooks 's wacky hotel comedy " The Nutt House . " The show , one of five new NBC series , is the second casualty of the three networks so far this fall .
MTL _{POS}	The NBC network canceled its first new series of the fall TV season , killing Mel Brooks 's wacky hotel comedy " The Nutt House . " The show , one of five new NBC series , is the second casualty of the three networks so far this fall .
STL	The NBC network canceled its first new series of the fall TV season , killing Mel Brooks 's wacky hotel comedy " The Nutt House . " The show , one of five new NBC series , is the second casualty of the three networks so far this fall .
MTL _{GRs} + ELMo	The NBC network canceled its first new series of the fall TV season , killing Mel Brooks 's wacky hotel comedy " The Nutt House . " The show , one of five new NBC series , is the second casualty of the three networks so far this fall .
MTL _{POS} + ELMo	The NBC network canceled its first new series of the fall TV season , killing Mel Brooks 's wacky hotel comedy " The Nutt House . " The show , one of five new NBC series , is the second casualty of the three networks so far this fall .
STL + ELMo	The NBC network canceled its first new series of the fall TV season , killing Mel Brooks 's wacky hotel comedy " The Nutt House . " The show , one of five new NBC series , is the second casualty of the three networks so far this fall .
MTL _{GRs} + BERT	The NBC network canceled its first new series of the fall TV season , killing Mel Brooks 's wacky hotel comedy " The Nutt House . " The show , one of five new NBC series , is the second casualty of the three networks so far this fall .
MTL _{POS} + BERT	The NBC network canceled its first new series of the fall TV season , killing Mel Brooks 's wacky hotel comedy " The Nutt House . " The show , one of five new NBC series , is the second casualty of the three networks so far this fall .
STL + BERT	The NBC network canceled its first new series of the fall TV season , killing Mel Brooks 's wacky hotel comedy " The Nutt House . " The show , one of five new NBC series , is the second casualty of the three networks so far this fall .

Figure 4.5: Visualisation of models’ attention weights on the WSJ dev set. Words that contribute the most to coherence scoring (i.e., those with high attention weights) are coloured: the contribution of words decreases from dark red to lighter tones of orange. I only colour the words that have weights higher than the median of the weights in their encompassing sentence.

‘Co.’ and in the second is ‘*company*’, both nominal subjects have dependent compounds: ‘*Ralston*’, ‘*Purina*’, ‘*St*’ and ‘*Louis*’ and both subjects are co-referential constituting a *continue* transition between the sentences.

Example (b), however, reveals a *retain* transition, where the direct object ‘*jetliners*’ is referred to by the subject ‘*aircraft*’ in the second sentence, and in that case the models still focus on the subjects and their compounds. The behaviour of attending to the subjects is not consistent though as can be seen in example (c) which also exhibits a *retain* transition and is ‘about’ a show by the NBC network. In this example, MTL_{GRs} does not focus on the subject in the first sentence, but highlights the name of the show (or part of it: ‘*Nutt*’) as well as ‘*hotel*’ that is part of the show’s description, then moves the attention to the subject ‘*show*’ in the second sentence, capturing the ‘aboutness’ of the text. On the same example, MTL_{POS} seems to capture less salient information according to Centering theory. As for the $MTL_{ELMo/BERT}$ approaches, they focus on the subject in the first sentence yet exhibit different behaviours in the second where $MTL_{POS+ELMo}$ and $MTL_{GRs+BERT}$ give a high weight to the subject ‘*show*’, whereas $MTL_{GRs+ELMo}$ and $MTL_{POS+BERT}$ ignore it.

It is also interesting that $MTL_{POS+ELMo}$ and $STL+ELMo$ tend to focus on verbs (i.e., *reported*, *earned*, *said*, *ordered* and *canceled*), which agrees with my quantitative results: they give the highest ranks to the root GR in Fig. 4.4 (b) and the VBD tag in Fig. 4.4 (d). As for the STL models, they appear to be more distracted than their MTL versions and less adhering to Centering theory. This postulates that learning syntactic properties of input words help make the models more focused. In general, we find that the models that achieve very similar performances in coherence modeling (Table 4.5) do not have identical focus points in text, yet they exhibit some similarity in the patterns they capture which potentially leads to similar predictions.

4.5 Realistic data experiments

In this section, I present the results of the three realistic GCDC datasets (§4.1.2) with further analysis in the second half of the section. I compare my models to the LC model¹⁷ and the PARSEQ one as they are the highest performing models by Lai and Tetreault (2018); I report their accuracies from Lai and Tetreault (2018).¹⁸ Nonetheless, I do not evaluate the EGrid or LCD models on the GCDC as they employ a pairwise ranking approach for optimisation and thus need modification to accommodate for the realistic domain (see §4.2.2 and §4.2.3).

¹⁷Lai and Tetreault (2018) refer to it as `CLIQUE` model.

¹⁸For the hyperparameters used for PARSEQ and LC, see Table 16 in ‘Supplementary Material’ in Lai and Tetreault (2018).

Model	Accuracy			
	Yahoo	Clinton	Enron	Avg.
LC	0.535	0.610	0.544	0.563
PARSEQ	0.549	0.602	0.532	0.561
STL	0.550	0.590	0.505	0.548
Concat _{GRs}	0.455	0.570	0.460	0.495
Concat _{POS}	0.470	0.555	0.455	0.493
MTL _{GRs}	0.560	0.620	0.560 *	0.580
MTL _{POS}	0.535	0.590	0.545*	0.556
MTL _{SOX}	0.505	0.585	0.510	0.533
MTL _{GRs+POS}	0.565	0.610	0.560 *	0.578
STL+ELMo	0.540	0.630*	0.525*	0.565
MTL _{GRs} +ELMo	0.565	0.610	0.540*	0.571
MTL _{POS} +ELMo	0.550	0.610	0.545*	0.568
STL+BERT	0.550	0.630*	0.550*	0.576
MTL _{GRs} +BERT	0.560	0.630*	0.525*	0.571
MTL _{POS} +BERT	0.565	0.635 *	0.535*	0.578

Table 4.6: Model accuracy on the three-way classification task on the GCDC. * indicates significance over STL with p-value < 0.01 using randomisation test. Avg. is the average accuracy for the three datasets. Results for PARSEQ and LC are those reported in Lai and Tetreault (2018) on the same data. Since Lai and Tetreault (2018) did not release their predictions for those models, I was unable to calculate significance for them.

4.5.1 Results

The results of the realistic data are reported in Table 4.6. We can see from the table that MTL_{GRs} achieves the best overall performance (0.580) based on the average accuracy of the three datasets. Furthermore, on individual datasets, we find that the highest accuracy is obtained by MTL_{POS}+BERT (Yahoo and Clinton) and MTL_{GRs+POS} (Yahoo and Enron). In general, the models that leverage a GR, POS or both auxiliary loss functions significantly outperform the STL baseline on at least one dataset. This further demonstrates that my MTL approach generalises to tasks involving the prediction of varying degrees of coherence in everyday writing. Moreover, we find that the models that leverage contextualised embeddings (STL or MTL) also significantly exceed the STL model on at least one dataset, further motivating the value of contextualised representations in various domains. With MTL, I managed to achieve state-of-the-art results on the GCDC, surpassing the best performing models published by Lai and Tetreault (2018): LC and PARSEQ.

Interestingly, we observe that MTL_{SOX} and the concatenation models (Concat_{GRs} and Concat_{POS}) do not generalise to the more realistic domain. Specifically, there is a substantial drop in the performance of the concatenation models compared to their performance on the WSJ. This could be attributed partly to the performance of the

Model	Low			Medium			High			Macro-F1		
	Yahoo	Clinton	Enron	Yahoo	Clinton	Enron	Yahoo	Clinton	Enron	Yahoo	Clinton	Enron
STL	0.643	0.250	0.444	0.0	0.0	0.0	0.576	0.731	0.643	0.406	0.327	0.362
MTL _{GRs}	0.642	0.441	0.586	0.0	0.0	0.235	0.603	0.743	0.653	0.415	0.395	0.491
MTL _{POS}	0.623	0.324	0.537	0.0	0.047	0.0	0.555	0.739	0.675	0.392	0.370	0.404
MTL _{GRs+POS}	0.632	0.379	0.578	0.0	0.010	0.0	0.626	0.747	0.675	0.419	0.409	0.417
STL+ELMo	0.629	0.509	0.526	0.0	0.0	0.068	0.561	0.769	0.640	0.396	0.426	0.411
MTL _{GRs} +ELMo	0.660	0.439	0.571	0.0	0.0	0.076	0.585	0.752	0.652	0.415	0.397	0.415
MTL _{POS} +ELMo	0.642	0.431	0.551	0.0	0.0	0.038	0.567	0.751	0.655	0.403	0.394	0.415
STL+BERT	0.628	0.494	0.505	0.0	0.0	0.037	0.597	0.761	0.674	0.408	0.418	0.405
MTL _{GRs} +BERT	0.655	0.484	0.477	0.0	0.0	0.0	0.594	0.775	0.658	0.416	0.420	0.378
MTL _{POS} +BERT	0.655	0.513	0.452	0.0	0.050	0.0	0.605	0.785	0.661	0.420	0.449	0.371

Table 4.7: The ‘Low’, ‘Medium’ and ‘High’ columns represent the F1-scores per class and the ‘Macro-F1’ represents the macro-averaged F1-score; scores are calculated across the GCDC datasets. The selected models are my best models based on the three-way classification accuracy reported in Table 4.6.

syntactic parser¹⁹ and partly to the nature of the GCDC dataset in terms of the properties of (in)coherence it exhibits compared to the WSJ articles. MTL gives the model more flexibility and control with respect to the features it learns (with the tuned α and β values) in order to enhance performance on the main task, in contrast to the concatenation models where the GRs and POS tags are forced directly as input to the model, yielding the worst performance across all the GCDC datasets.

Although the evaluation metric employed on realistic data is different from the binary task, we notice that the numbers obtained on this dataset are quite low compared to those on the WSJ. Assessing varying degrees of coherence is a more challenging task; the discrepancy in coherence between different documents is less pronounced than when randomly shuffling sentences in a coherent document. This is also shown by the ‘fair’ human agreement reported in Table 4.2 which indicates the subjectivity of judging coherence in everyday writing. In addition, the GCDC is a noisy domain that includes grammatical and spelling mistakes and different styles of writing as could be seen in Appendix A. This makes the syntactic parsers more susceptible to committing errors which might negatively affect the models that use syntactic features, particularly if the features are concatenated with input word vectors. Finally, the GCDC datasets are small in size (900 documents for training distributed on three classes), resulting in a low representation for each class which makes it harder for the models to learn useful discourse-related features. I next investigate the performance per class and further analyse the realistic data results.

4.5.2 Analysis

4.5.2.1 Performance per class

I report in Table 4.7 the F1-scores for the three coherence classes: low, medium and high. I also report the averaged macro-F1 score calculated as the arithmetic mean of the per-class F1-scores. Investigating individual class performance is useful to get a closer look into the behaviour of the models and examine where they fall short. The striking result revealed by Table 4.7 is that the models fail to assess documents of medium coherence, which can be explained by the small number of training examples representing this class, in comparison to the other classes (Table 4.2). Additionally, assessing texts of medium coherence is challenging and it is even hard for expert annotators to reach “acceptable levels of agreement” on this middle class (Burstein et al., 2013). I further investigate the ability of the models to capture different coherent levels in §4.5.2.3.

4.5.2.2 Transfer learning

I employ transfer learning to tackle the data scarcity problem in the GCDC domain; I particularly leverage *sequential adaptation* where a model is trained first on a related source task then fine-tuned on the target task (Mou et al., 2016; Min et al., 2017; Chung et al., 2018). I use the WSJ as the source domain and the GCDC as the target one. This choice of source domain is motivated by the similarity of the source and target tasks (both are coherence assessment), and the high performance obtained by the different MTL and ELMo/BERT-based approaches on the WSJ. Furthermore, the sensitivity of the models to sentence order (§4.4.3.1) suggests that leveraging the parameters learned from this source domain might provide a further boost in the target domain. I apply two main steps for transfer learning: (1) training a neural network on the WSJ binary domain and (2) using the resulting pre-trained parameters to initialise the GCDC network and fine-tune it to perform the coherence multi-class prediction task. For simplicity, I exclude from the parameter transfer any layer that has a different size between the two domains. This includes the input word embedding layer and its associated Bi-LSTM layer (Eq. 3.1), the final prediction layer (Eq. 3.7) and the layers associated with paragraph representations since they do not exist in the WSJ model (Eqs. 3.3 and 3.4). Accordingly, the transferred weights are the ones associated with building sentence and document representations (Eqs. 3.2, 3.5 and 3.6). I apply transfer learning to the model that achieves the highest average accuracy: MTL_{GR_s} and report the results in Table 4.8.

From Table 4.8, we can see that transfer learning from the WSJ domain to the GCDC one does not help. The three-way classification accuracy drops on Yahoo and Enron

¹⁹Parsers are traditionally trained on the Penn Treebank and thus can be more accurate with datasets like the WSJ.

Model	Three-way Classification Accuracy for All Classes			F1-scores											
				Low			Medium			High			Macro-F1		
	Yahoo	Clinton	Enron	Yahoo	Clinton	Enron	Yahoo	Clinton	Enron	Yahoo	Clinton	Enron	Yahoo	Clinton	Enron
MTL _{GRs}	0.560	0.620	0.560	0.642	0.441	0.586	0.0	0.0	0.235	0.603	0.743	0.653	0.415	0.395	0.491
MTL _{GRs} +TL	0.545	0.630	0.520	0.631	0.470	0.487	0.0	0.051	0.0	0.573	0.760	0.649	0.401	0.427	0.378

Table 4.8: A comparison between MTL_{GRs} with no pre-training and when it is pre-trained on the WSJ then fine-tuned on GCDC (MTL_{GRs}+TL (transfer learning)). The ‘Three-way Classification’ column displays the overall three-way classification accuracy for each dataset, similar to Table 4.6, while the rest of the columns detail the F1-scores for each class and macro-averaged F1-score similar to Table 4.7.

texts, while there is a minor 1% increase on Clinton emails. A similar result is observed from the F1-scores that denote the per class performance. I ascribe that to the different idiosyncrasies of the two domains in terms of coherence features and style. The incoherent documents in the WSJ are synthetically created and thus exhibit rough entity shifts, whereas the (in)coherence features in everyday writing are less pronounced. Moreover, the WSJ is a formal domain with a style specific to news articles, while the GCDC is less formal and less constrained in style. In the future, I would like to investigate more transfer learning approaches such as adaptive learning rates (Howard and Ruder, 2018), or partially fine-tuning the pre-trained network (Chung et al., 2018), instead of fine-tuning all the transferred parameters. I would also like to conduct cross-domain experiments between the different GCDC domains.

4.5.2.3 Ranking coherence

Three-way classification accuracy, that I have used so far, might not well-represent the levels of coherence that exist in the data, especially that the gold labels are the consensus labels that average the raters’ scores and use thresholding to map the resulting mean score to a coherence class (§4.1.2). To remedy this, I follow another evaluation setup by Lai and Tetreault (2018) and measure the Spearman’s rank correlation coefficient (ρ) between the predicted scores and the gold ones, calculated by averaging the raters’ scores with no thresholding. The resulting gold labels are more fine-grained, which allows us to address the low representation of the medium class in the corpus. In this experimental setup, the network hyperparameters and architecture stays the same, except for the output layer as it should predict one label via regression, instead of three. To that end, Eq. 3.7 changes to:

$$\hat{y} = W^d h^d \quad (4.8)$$

where $W^d \in \mathbb{R}^{dim^d \times 1}$ and dim^d is the length of h^d . The loss function optimises the MSE following Eq. 3.9 but only between the gold score and the predicted one, instead of three classes. The best model I leverage for testing is the one that achieves the highest Spearman’s correlation on the dev set. For this experiment, I test the strongest models

Model	Spearman (ρ)			
	Yahoo	Clinton	Enron	Avg.
LC	0.474	0.474	0.416	0.454
PARSEQ	0.519	0.448	0.454	0.473
STL	0.435	0.399	0.444	0.426
MTL _{GRs}	0.445	0.409	0.453	0.435
MTL _{POS}	0.445	0.423	0.459	0.442
MTL _{GRs+POS}	0.443	0.416	0.456	0.438
STL+ELMo	0.508	0.496	0.519	0.507
MTL _{GRs} +ELMo	0.465	0.528	0.526	0.506
MTL _{POS} +ELMo	0.460	0.539	0.479	0.492
STL+BERT	0.551	0.562	0.550	0.554
MTL _{GRs} +BERT	0.524	0.606	0.504	0.544
MTL _{POS} +BERT	0.522	0.586	0.502	0.536

Table 4.9: Spearman’s rank correlation coefficient (ρ) on the GCDC. Avg. is the average ρ for the three datasets; applying Fisher transformation (§2.7) had no/very negligible effect so I simply take the average. Results for PARSEQ and LC are those reported by Lai and Tetreault (2018) on the same data.

on the three-way classification task according to Table 4.6. I calculate the Spearman’s correlation between the predicted labels and the ground-truth ones and report the results in Table 4.9. The results show that, overall, BERT-based models have the highest predictive power, followed by ELMo-based models. Unlike the WSJ domain, the difference between ELMo and BERT is more notable and their encoded features are better manifested in this task. In general, the results of the contextualised-based models on classification (Table 4.6) as well as ranking (Table 4.9) further demonstrate their ability to capture varying degrees of coherence, in both STL and MTL frameworks. Nonetheless, despite the relatively high performance of the MTL_{GRs} and MTL_{GRs+POS} models on the classification task, their performance drops on the ranking task, in comparison to the other models. This discrepancy between the two tasks motivates further research to explore the methods utilised to annotate coherence data and the metrics used for evaluation.

4.5.2.4 Attention visualisation

Similar to my analysis in §4.4.3.3, I visualise the attention weights of the models trained on Yahoo posts. I select Yahoo for my analysis as it has the highest inter-rater agreement (Table 4.2).

Quantitative analysis. I calculate the importance and rank scores for GRs and POS tags and plot them in Fig. 4.6. First, I analyse the attention scores for the GR labels (Fig. 4.6 (a) and (b)). We find that, similar to the WSJ, MTL_{GRs} and MTL_{POS} give the highest focus to the subject role. However, they also give high attention to less salient roles

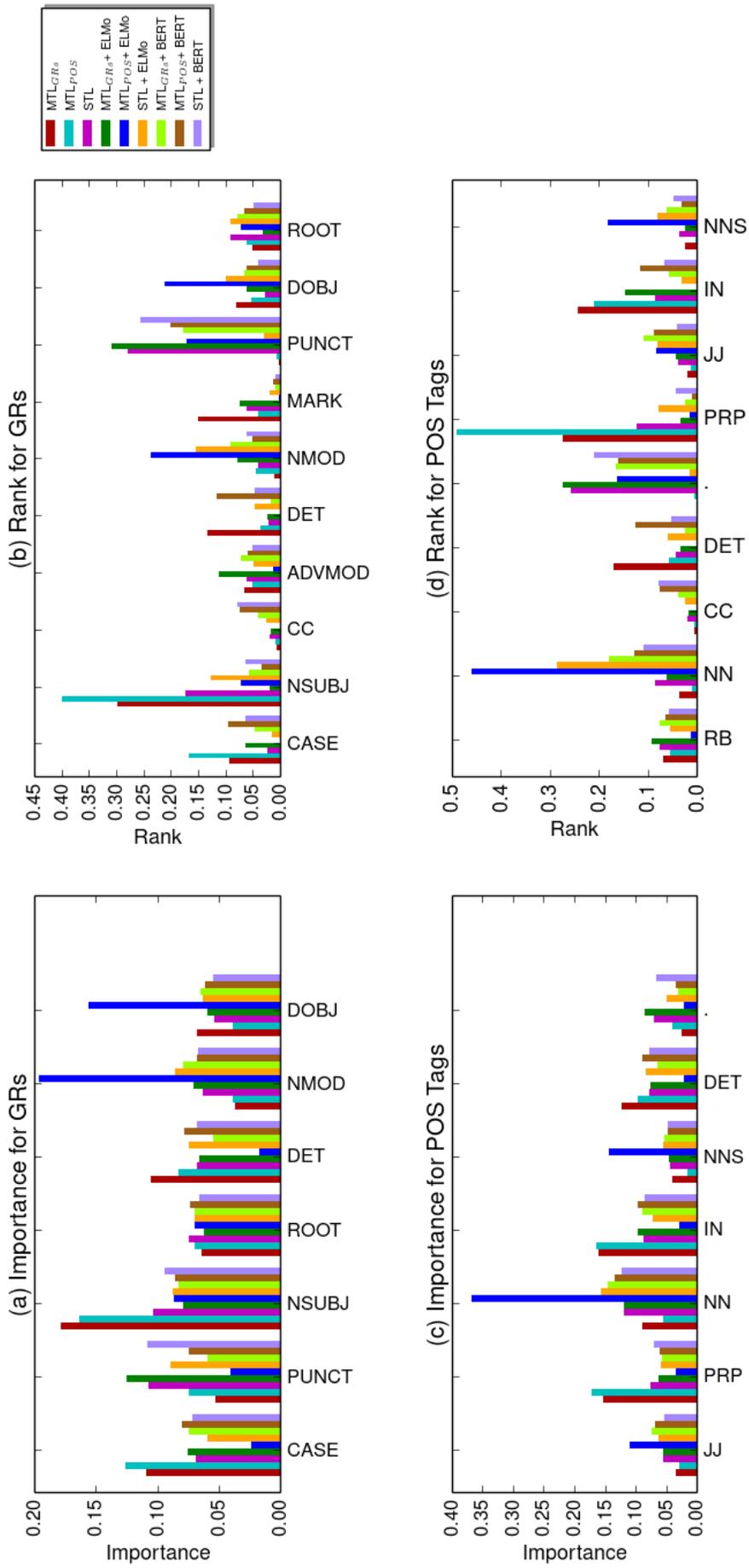


Figure 4.6: Attention analysis on Yahoo dev set. GRs or POS tags are displayed on the x-axis and their corresponding importance and rank scores on the y-axis.

such as the *case* label, as revealed in Fig. 4.6 (a). On the other hand, STL seems to give distracted importance scores, but gives the highest rank the *punct* role. A similar behaviour is also observed with the three BERT models. As mentioned in §4.4.3.3, punctuation marks can play an important role in discourse structure which could explicate why many contextualised models attend to them. Regarding ELMo models, MTL_{GRs}+ELMo focuses on punctuation marks, whereas MTL_{POS}+ELMo assigns more weight to nominal modifiers (*nmod*)²⁰ and direct objects (*obj*), and STL+ELMo does not exhibit any striking attention patterns.

As for the POS scores (Fig. 4.6 (c) and (d)), we find that the high attention given by MTL_{GRs} and MTL_{POS} to subject and case GRs, translates to high attention to personal pronouns (*PRP*) and preposition/subordinating conjunction (*IN*) tags. On the other hand, the STL model and BERT models are less focused, with the first giving a high rank score to the “.” tag. Regarding ELMo models, MTL_{GRs}+ELMo also seems less focused and assigns the highest rank to the “.” tag, whereas MTL_{POS}+ELMo and STL+ELMo (more notably the first) attends to the *NN* tag.

Qualitative analysis. I further conduct a qualitative analysis and display in Fig. 4.7 the attention weights of the first three sentences from a document, from Yahoo dev set, that is annotated with a high coherence class. The analysis reveals that, in contrast to the WSJ, the highlighted words do not exhibit specific patterns and thus it is harder to tell what coherence features are captured. For instance, MTL_{GRs} on the WSJ tends to focus more on nouns specifically subjects and their associated compounds (Fig. 4.5), while in the example in Fig. 4.7 the highest attention is assigned to prepositions and determiners. The only interesting observation is that STL+ELMo and STL+BERT highlight ‘*bench warrant*’ in the first and third sentences, which captures the ‘aboutness’ of text.

In general, the quantitative and qualitative analysis on Yahoo posts postulates the premise that with realistic data, coherence features are less pronounced and it is hard to associate them with what the models focus on. Additionally, as mentioned earlier, the WSJ articles exhibit some regularities in style that the models can pick up on, whereas everyday writing has a more free style and thereby it is more challenging to identify its coherence properties. The small size of the dataset also adds to this challenge as there are not enough features to learn from. Attention visualisation could be more helpful in a constrained synthetic domain than in a more realistic one. Furthermore, the attention given by many models to punctuation marks motivates further investigation to explicate the role they play in discriminating between texts of different coherence levels, which is an interesting direction for future work. I finally note that in texts written by non-professional writers such as online posts or emails, syntactic parsers are more susceptible to committing

²⁰E.g., ‘*nature*’ in “Everything follows from the true nature of things.”

errors which, again, makes the visualisation analysis less informative than the formal WSJ domain.

MTL _{GRs}	A bench warrant is a order given by the judge that presides over a case to law enforcement officers . The order is to find and arrest a suspect at first sight . A bench warrant is and can be sworn at anytime prior to or during a trial .
MTL _{POS}	A bench warrant is a order given by the judge that presides over a case to law enforcement officers . The order is to find and arrest a suspect at first sight . A bench warrant is and can be sworn at anytime prior to or during a trial .
STL	A bench warrant is a order given by the judge that presides over a case to law enforcement officers . The order is to find and arrest a suspect at first sight . A bench warrant is and can be sworn at anytime prior to or during a trial .
MTL _{GRs} + ELMo	A bench warrant is a order given by the judge that presides over a case to law enforcement officers . The order is to find and arrest a suspect at first sight . A bench warrant is and can be sworn at anytime prior to or during a trial .
MTL _{POS} + ELMo	A bench warrant is a order given by the judge that presides over a case to law enforcement officers . The order is to find and arrest a suspect at first sight . A bench warrant is and can be sworn at anytime prior to or during a trial .
STL + ELMo	A bench warrant is a order given by the judge that presides over a case to law enforcement officers . The order is to find and arrest a suspect at first sight . A bench warrant is and can be sworn at anytime prior to or during a trial .
MTL _{GRs} + BERT	A bench warrant is a order given by the judge that presides over a case to law enforcement officers . The order is to find and arrest a suspect at first sight . A bench warrant is and can be sworn at anytime prior to or during a trial .
MTL _{POS} + BERT	A bench warrant is a order given by the judge that presides over a case to law enforcement officers . The order is to find and arrest a suspect at first sight . A bench warrant is and can be sworn at anytime prior to or during a trial .
STL + BERT	A bench warrant is a order given by the judge that presides over a case to law enforcement officers . The order is to find and arrest a suspect at first sight . A bench warrant is and can be sworn at anytime prior to or during a trial .

Figure 4.7: Visualisation of models’ attention weights on Yahoo dev set. Words that contribute the most to coherence scoring (i.e., those with high attention weights) are coloured: the contribution of words decreases from dark red to lighter tones of orange. I only colour the words that have weights higher than the median of the weights in their encompassing sentence.

4.6 Summary

In this chapter, I have compared my MTL approach (that utilises GRs or POS tags) to other strong neural benchmarks including: a model with the same architecture that only performs the single task of coherence scoring (STL), a local coherence model (LC), models that feed syntactic properties as input to the network (Concat models) and models that incorporate two auxiliary functions to learn both GR and POS labels (MTL_{GRs+POS}). Furthermore, I enhanced my STL and MTL approaches with contextualised word embeddings (ELMo or BERT) and compared them to standard pre-trained embeddings. I applied my experiments to two domains of coherence assessment: a binary domain where the model ranks a well-

written document against its noisy counterparts constructed by shuffling the sentences in the original document, and a realistic domain that consists of texts representing everyday writing with various coherence degrees.

As for the binary domain, I compared my approaches to existing state-of-the-art models trained in a pairwise fashion that either leverage entity grids in a neural framework (CNN-EGrid_{ext} and CNN-EGrid_{lex}) or apply a local discrimination approach that either has a generative backend model (LCD-L) or average word representations (LCD with fastText, ELMo or BERT). My results showed the efficacy of MTL, particularly using contextualised embeddings, and the power of the LCD approach, also using contextualised embeddings. My MTL approach with BERT or ELMo achieves state-of-the-art TPRA in binary coherence assessment, while LCD-BERT yields state-of-the-art PRA on the same task. The results also showed that utilising the whole set of GRs (MTL_{GRs}) as auxiliary labels outperforms only focusing on subject and object roles (MTL_{SOX}), and that leveraging both GRs and POS tags together (MTL_{GRs+POS}) does not have an advantage over using them separately.

Regarding the realistic data domain, the best overall performance is obtained by the GR-based models (MTL_{GRs} and MTL_{GRs+POS}) when the task is cast as multi-class classification. However, we observe that using BERT embeddings (with MTL or STL) yields the strongest and most consistent performance on multi-class classification or ranking where the task becomes a regression problem evaluated with Spearman’s rank correlation coefficient. Furthermore, we find that all the models fail to capture documents of medium coherence level.

I backed my results with further analysis and investigated the features the models focus on. My analysis revealed some consistent patterns on the synthetic data as we find that the MTL-based models tend to focus on words that appear as subjects, further corroborating Centering theory of coherence. On the other hand, in the realistic domain, analysing the attention weights was less insightful. The general discrepancy of the performance in the synthetic domain vs. the realistic one as well as the findings revealed by attention visualisation can be attributed to the nature of the two domains. Synthetic data exhibits rough shifts and distorted syntactic patterns while (in)coherence features are less pronounced in realistic data. Additionally, the size of the realistic training data and the low representation of coherence classes contribute to the drop in performance. This warrants further research to curate more representative data and investigate the utilised annotation criteria.

EVALUATION OF DISCOURSE COHERENCE

As previously discussed in Chapter 2, different theories have been proposed to describe the properties that contribute to discourse coherence and some have been translated to computational models for empirical evaluation such as entity-based approaches (Barzilay and Lapata, 2008; Filippova and Strube, 2007; Burstein et al., 2010; Elsner and Charniak, 2011b; Guinaudeau and Strube, 2013), probabilistic models that focus on syntactic patterns (Louis and Nenkova, 2012) or topic shifts (Barzilay and Lee, 2004), modeling rhetorical relations (Mann and Thompson, 1988; Lin et al., 2011; Feng et al., 2014) and capturing semantic relatedness between sentences (Lapata and Barzilay, 2005; Soricut and Marcu, 2006; Somasundaran et al., 2014). More recently, neural networks have been popular in coherence modeling either by leveraging EGrid representations of text (Tien Nguyen and Joty, 2017; Joty et al., 2018) or automatically learning useful representations in an end-to-end fashion (Li and Jurafsky, 2017; Logeswaran et al., 2018; Farag and Yannakoudakis, 2019; Xu et al., 2019; Moon et al., 2019). In Chapter 4, I have shown that my MTL approach and other state-of-the-art coherence models can efficiently discriminate between coherent and incoherent texts, particularly when enhanced with contextualised embeddings. I have also attempted to interpret model performance with visualisation (§4.4.3.3 and §4.5.2.4) to investigate the salient features they focus on. Nonetheless, in a complex task like coherence assessment, interpretability is challenging as there are many contributing textual and inferential factors, which motivated me to conduct further investigation to understand the linguistic features captured by the models.

The high performance obtained by previous neural approaches, in addition to my models, has rendered the coherence binary discrimination task solved. As a result, work on coherence modeling has focused on more challenging tasks such as recovering the correct sentence order (Logeswaran et al., 2018; Cui et al., 2018; Oh et al., 2019), evaluation on realistic data (Lai and Tetreault, 2018; Farag and Yannakoudakis, 2019) and open-domain models of coherence (Li and Jurafsky, 2017; Xu et al., 2019). However, less attention has been directed to investigating what the models actually learn, which is needed to

provide insight into how to frame the task and improve the models. Li and Jurafsky (2017) conducted a qualitative analysis by testing their models on a few examples that exhibit different coherence features, with the aim of establishing a direction for future research that examines the strengths and shortcomings of coherence models. This inspired me to create an evaluation framework to investigate the linguistic features learned by coherence models. I hypothesise that the models are able to capture certain syntactic patterns that occur in coherent documents, while failing to identify other semantic and topical aspects of text. This idea is motivated by the work of Louis and Nenkova (2012) who analysed the syntactic patterns that co-occur in the WSJ, as will be discussed in §5.1.

In this chapter, I present an evaluation framework for systematically investigating how well current models of coherence can capture aspects of text implicated in discourse organisation. I analyse the ability of the models to capture certain inter-sentential properties with respect to model architecture and pre-training domain. My evaluation framework consists of two main evaluation tasks:

- I compile a large-scale dataset on which I apply syntactic and semantic perturbations and test the ability of coherence models to detect them. I leverage this dataset by: (1) directly evaluating the pre-trained coherence models on the test portion of the dataset or (2) using the latent representations of the pre-trained coherence models to train a classifier and then evaluate it on the test split of the data, in order to adapt the models to the new test domain.
- I carefully create a more-controlled smaller-scale dataset from the news domain, similar to the WSJ, and conduct an error analysis to systematically assess the sensitivity of the models to changes in relevant syntactic or semantic patterns.

I evaluate a wide range of state-of-the-art neural approaches pre-trained on standard synthetic data (the WSJ). I want my choice for the pre-trained models to cover high performing approaches that exhibit architectural and algorithmic diversity. Therefore, I evaluate the 6 main MTL models (MTL_{GRs}, MTL_{POS}, MTL_{GRs}+ELMo, MTL_{POS}+ELMo, MTL_{GRs}+BERT and MTL_{POS}+BERT), the 4 LCD models (LCD-L, LCD-fastText, LCD-ELMo and LCD-BERT), the STL models (STL, STL+ELMo and STL+BERT), the best performing EGrid model (EGrid CNN_{ext}) and the LC model. I further extend my experiments to models pre-trained on the Yahoo dataset to investigate the impact of pre-training on a realistic domain. I note that this chapter is based on a long paper published in the 1st Workshop on Computational Approaches to Discourse (CODI 2020) (Farag et al., 2020).¹ I hope that this work will provide a platform for coherence evaluation and will be extended to examine more coherence-related phenomena.

¹The datasets presented in this chapter are available at <https://github.com/Youmna-H/coherence-analysis>.

I note that Chen et al. (2019) presented a set of discourse-aware tasks to test the ability of sentence encoders to capture the surrounding context of sentences. The tasks included predicting a sentence position and whether a sentence pair is in the correct order. They also utilised discourse-aware training objectives such as predicting neighbouring sentences given a sentence and the title of the section or document the sentence belongs to. My work differs from theirs in that I focus on evaluating models that are built and trained to capture discourse coherence rather than general purpose sentence encoders. I also focus on semantic and syntactic inter-sentential properties that I further control in my small-scale test set to help pinpoint the features captured by the models.

The rest of the chapter is organised as follows. In §5.1, I explain the syntactic structure of text and its impact on the organization of discourse. In §5.2, I present the large-scale dataset, and discuss the performance of the coherence models when directly evaluated on it, or when further fine-tuned. In §5.3, I introduce the controlled small-scale test set, detail how its examples are crafted and discuss how the models perform on these examples. Finally, in §5.4, I conclude the chapter by summarising my findings.

5.1 Syntactic structure

Syntax, or syntactic structure, refers to word order and how words and phrases are arranged to form sentences in a language. Grammatical productions define a finite set of rules that describes the syntax of a language and are used to generate all possible sentences in that language. For instance, the sentence: “Mary likes reading” can be generated by the rules:

S -> NP VP	NNP -> Mary
NP -> NNP	VBZ -> likes
VP -> V NP	NN -> reading
V -> VBZ	
NP -> NN	

A battery of data-driven studies have investigated the exhibition and influence of syntactic patterns in discourse. Some research has focused on syntactic consistency in spoken dialogues (Reitter et al., 2006; Pietsch et al., 2012), while other examined the syntactic repetitions in corpus data such as the WSJ (Dubey et al., 2005; Cheung and Penn, 2010b). They all showed that syntactic structures tend to be reused in consecutive utterances/sentences. In contrast to these studies that focus on the repetition of syntactic constructions, Louis and Nenkova (2012) examined whether different syntactic structures tend to appear together in adjacent sentences. They analyzed the grammatical production rules that co-occur in consecutive sentences in Section 0 of the WSJ in a total of 1,727 sentence pairs. They found that there are 197 unique productions

forming 38,809 production pairs among which 1,168 appear significantly more often than chance and 172 occur significantly less. Their analysis revealed that the pair $S \rightarrow NP\text{-}SBJ\ VP \mid NP\text{-}SBJ \rightarrow PRP$ is the most frequent one with 290 occurrences, where the first sentence introduces a subject and predicate and the subject is pronominalised in the second sentence. They also found that some of the co-occurring productions involve numbers and quantities which could be attributed to the financial nature of the WSJ.

As discussed in §2.2.2, the syntax of a sentence could be used as a proxy for its communicative goal and each sentence type has distinguishable syntax (e.g., questions or definitions). This is further demonstrated by Cocco et al. (2011) who discovered a significant relationship between the POS tags that appear in a sentence and its linguistic type such as narration, dialogue, or explanation in French short stories. As the syntax of a sentence plays a key role in conveying its communicative goal and, in turn, the intentional structure of discourse, a sequence of sentences in a coherent text is expected to exhibit syntactic regularities. This might suggest that models memorise that certain syntactic patterns occur in coherent documents, and thus are able to distinguish them from incoherent permuted versions where those patterns are broken. While syntax is important in coherence modeling, semantics also plays an inherent role that should not be overlooked by the models. For example, the following sentence pairs both have the same syntactic structure but the choice of words makes (a) more plausible and coherent.

- (a) Mary likes reading. She buys many books.
- (b) Mary likes reading. She eats many cupcakes.

RNNs and syntax RNNs have been a popular approach to capture syntax, particularly their enhanced variations, such as LSTMs, that are more capable of modeling long-distance grammatical dependencies. Many efforts have been devoted to test the ability of RNNs to learn syntactic structures (Linzen et al., 2016; Blevins et al., 2018; Gulordava et al., 2018; Wilcox et al., 2019). For instance, in the task of predicting subject-verb number agreement (Bock and Miller, 1991), where models need to capture syntactic dependencies, Linzen et al. (2016) found that LSTMs attain promising results even with multiple words (*attractors*) occurring between the verb and its subject.² On the same task, Kuncoro et al. (2018) showed that the performance of LSTMs can be significantly enhanced by increasing their capacity to address the problem of multiple attractors. More interestingly, they found that there are no performance gains from providing LSTMs with syntactic annotations (e.g., phrase-structure tree annotations). Gulordava et al. (2018) demonstrated that an RNN language model is capable of capturing long-distance number agreement even in nonsensical sentences where a sentence is grammatical but does not make sense (e.g., the popular linguistics sentence “Colorless green ideas sleep furiously.”

²Error rates, however, get higher when increasing the number of attractors.

(Chomsky, 1957, p. 15)). They, thereby, showed that RNNs can learn syntactic structures with no recourse to semantic or lexical cues.

On the other hand, Marvin and Linzen (2018) showed that LSTMs are very effective in capturing simple structures for subject-verb number agreement, yet they fail to detect more complex constructions, indicating that their performance is dependent on the nature of the leveraged data. Additionally, other models were proven to be more capable of capturing subject-verb agreement such as recurrent neural network grammars (RNNGs) (Dyer et al., 2016; Kuncoro et al., 2018) or BERT (Goldberg, 2019), showing that there is more room for improvement over sequential RNNs.

All this shows that RNN-based architectures are able to encapsulate syntactic information, but they have their limitations. I conjecture that in a dataset like the WSJ, even with its complex structures, the repetition of syntactic patterns might give the models a straightforward signal to discriminate between the intact patterns and the shuffled ones. However, this needs empirical investigation as will be detailed in the rest of this chapter.

5.2 Cloze Coherence Dataset

I devise a *large-scale* dataset of coherent and incoherent instances, where the coherent examples are intact well-written texts and the incoherent ones are the result of applying syntactic or semantic perturbations to the coherent examples. I refer to this dataset as the Cloze Coherence Dataset (CCD). I first start by explaining how the coherent instances are created then move on to the incoherent ones.

5.2.1 Coherent examples

Ideally, I want the coherent cases to consist of two short sentences³ that are coreferential and exhibit a rhetorical relation, where these properties are removed in the incoherent counterparts. Furthermore, the examples should be self-contained, meaning that they do not reference or rely on any outer textual context to be interpreted. I find that narrative texts are good candidates to satisfy this criteria, and hence, create my coherent examples from the ROCStories Cloze dataset (Mostafazadeh et al., 2016). This dataset contains short stories of 5 sentences, written by Amazon Mechanical Turk workers and exhibit a sequence of causal or temporal events that have a shared protagonist. A story usually starts by introducing a protagonist in the first sentence, then subsequent sentences describe events that happen to them in a logical / rhetorically plausible manner. The dataset was designed for common-sense reasoning by testing the ability of machine learning models to select a plausible ending for the story out of two alternative endings. Here, my goal is not

³I want to specifically test for coherence so I avoid complex linguistic structures.

Type of reference	Example
Pronominal Reference	Rich was a musician. <u>He</u> made a few hit songs.
Proper Name	Dan’s parents were overweight. <u>Dan</u> was overweight as well.
Nominal Substitution	My dog hates his treats. I decided to go buy some new <u>ones</u> .
Demonstrative Reference	My daughter wants to take her toddler to the Enchanted Village. <u>This</u> is a puppet show featuring early 20th century figurines.

Table 5.1: Examples of first two sentences extracted from the ROCStories Cloze dataset to demonstrate different reference types, with the referring word underlined.

to challenge the models to select the right ending but rather to capture inter-sentential relations and coherence-related features. That is why I only select the first two sentences in the stories to compose the coherent examples in my dataset.

I first investigate the validity of creating the coherent instances from the first two sentences and whether those instances exhibit the aforementioned properties of being self-contained, rhetorically related and coreferential. Selecting the first two sentences should result in self-contained examples since there is no preceding context they refer to, and no cataphoric relations to consequent sentences. As for the rhetorical relations, Mostafazadeh et al. (2016) conducted a *temporal analysis* to investigate the logical order of the events presented in a story. They created two datasets: the first by selecting 50 *good stories* written by the top workers and the second by randomly choosing other 50 stories from the dataset (*random stories*). They then shuffled the sentences in each story in the two sets, and asked five annotators to reorder them. They took the majority ordering of the five crowd workers for each story and found that for the good stories, 100% of the majority ordering agrees with the original order and with the random stories the percentage becomes 86%. More specifically, they found that the annotators place the first sentence in its correct position 98.8% of the times in the good stories and 95.6% in the random ones, whereas the percentage of correctly placing the second sentence in the good stories is 97.6% and in the random ones 86%. These percentages suggest that the stories are presented in a commonsensical temporal manner with logical links between consecutive sentences, further strengthening the validity of this dataset.

As for the coreferential relations between the two sentences in each extracted pair, I examine them by gathering some statistics. I initially used spaCy (Honnibal and Johnson, 2015) and the Stanford coreference resolution system (Clark and Manning, 2016) to find if both sentences contain mentions of the same entity, but found their performance unreliable for the purposes of this experiment after manual inspection. Therefore, I adopt a heuristic approach by simply counting the number of second sentences that contain at least one third person pronoun (either personal or possessive) and find that they constitute 80% of the examples. Third person pronouns anaphorically refers to preceding items in text, which could occur in the same sentence or the previous one (i.e., the first sentence). I, therefore, randomly select, and manually inspect, 500 examples that contain third person pronouns

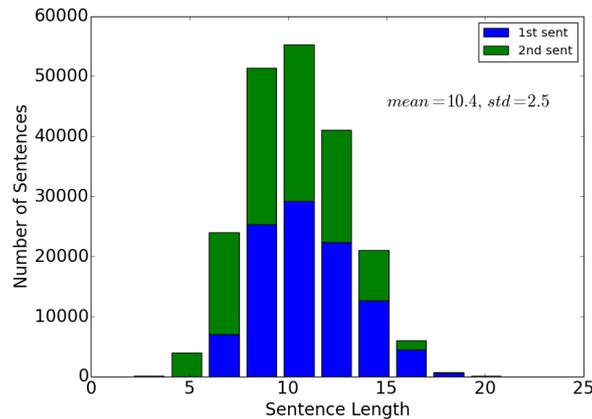


Figure 5.1: The distribution of sentence length in the coherent examples in the CCD. The blue colour refers to the first sentence and the green to the second one in the examples. The average sentence length, regardless of its position = 10.4 with standard deviation (std) = 2.5.

in their second sentence and find that in 95% of them the referenced entity appears in the first sentence. Furthermore, third person pronouns are not the only coreferential relations in the examples. For instance, I find that 90% of the second sentences contain a personal or possessive pronoun (whether it is first, second or third person), which could also signal coreference, e.g., ‘I was walking to school. Since I wasn’t looking at my feet I stepped on a rock.’ There are also other coreferential devices such as: demonstrative references (e.g., ‘this’ and ‘there’), ‘the’ + noun, proper names or nominal substitutions (e.g., ‘one’ or ‘ones’) to name a few (§2.1.1), so the true proportion of coreferential pairs will be higher. This further strengthens my hypothesis that the first two sentences are coherently tied up, thus constituting good examples for my experiments. Table 5.1 shows examples of different referential relations in the dataset. The examples in the table reveal cohesive ties between the sentences in addition to semantic relations (e.g., in the third example there is a causal relation between the two sentences).

For the train/dev/test splits, I follow the same division as Mostafazadeh et al. (2016); I exclude some examples with erroneous sentence boundaries,⁴ yielding 97,903 examples for training, 1,871 for development, and 1,871 for testing, with training vocabulary size = 29,596 tokens. Each example contains two sentences that represent a coherent pair. Figure 5.1 depicts the sentence length distribution in my extracted dataset.

5.2.2 Incoherent examples

I want to create the incoherent examples in a way that would test the susceptibility of the models to syntactic or semantic alterations. To that end, I follow one of two methods:

⁴The training stories are in CSV format (separating sentences by comma delimiters) and I parse them using the Python CSV parser. I exclude the stories where the parser fails to detect 5 sentences.

(1) corrupt the syntactic pattern in each coherent pair (2) change the semantic relation between the sentences. This results in two datasets explained as follows.

cloze_swap I swap the two sentences in each pair which mostly breaks the coreference relation between them, and/or breaks the rhetorical relation (e.g., temporal or causal) by reversing the event sequence. I refer to this dataset as **cloze_swap**. This dataset constitutes examples of corrupted syntactic patterns; i.e., affecting the grammatical productions that co-occur in coherent texts (§5.1). Additionally, it is a balanced dataset of coherent and incoherent examples since the incoherent instances are simply created by swapping the sentences in the coherent ones.

cloze_rand The aim of this dataset is to change the semantic relation between the two sentences to produce an incoherent pair. To that end, I keep the first sentence intact and randomly choose a second unrelated sentence from the dataset.⁵ I refer to this dataset as **cloze_rand**. For each coherent pair, I compose one incoherent random version to keep the data balanced as with **cloze_swap**. One problem that might arise is that the randomly-created pair might still be coherent, so I address this by the following:

- I find that around 70% of the second sentences in the corpus *start* with a pronoun (either personal or possessive), which as mentioned earlier refers to an entity in the previous sentence. I, therefore, constrain the random selection of the second sentence so that it does not start with the same word as the second sentence in the original pair. I also extend this constraint and do not select a sentence that starts with the pronoun ‘he’ if the original starts with ‘she’ and vice-versa.⁶
- The previous point describes a heuristic way of creating the incoherent pairs in which I attempt to break the continuity between the two sentences to make the resulting pair less coherent than the original intact one. While the random selection would most likely result in an implausible pair and applying the previous constraint will further help with that, it is still not guaranteed that the incoherent pair will be less coherent than the original one. That is why I use human evaluation to further assess the validity of the dataset in addition to setting the upper-bound performance on the task. I randomly select 100 coherent examples from the test set along with their randomly-generated counterparts and ask two annotators, with high English proficiency levels, to rank them based on which pair they think is more coherent. The annotators were particularly instructed to “rank each pair of examples according

⁵This is similar to the next sentence prediction task on which BERT is pre-trained (Devlin et al., 2019). Here, the random sentence is selected only from the second sentences of the other pairs in the dataset, and not from the sentences in other positions.

⁶I do not find instances of ‘they’ used as a third-person singular pronoun.

Coherent example	Incoherent example from cloze_swap	Incoherent example from cloze_rand
Tyrese joined a new gym. The membership allows him to work out for a year.	The membership allows him to work out for a year. Tyrese joined a new gym.	Tyrese joined a new gym. As children they hated being dressed alike.
Jasmine doesn't know how to play the guitar. She asked her dad to take her to guitar class.	She asked her dad to take her to guitar class. Jasmine doesn't know how to play the guitar.	Jasmine doesn't know how to play the guitar. May thought her milk was no good.
I wanted to play an old game one day. When I looked in the game's case the CD was missing.	When I looked in the game's case the CD was missing. I wanted to play an old game one day.	I wanted to play an old game one day. Jason pressed the buzzer since he knew the answer.

Table 5.2: Coherent and incoherent pairs from the cloze_swap and cloze_rand datasets.

to what they see as more coherent, plausible and could be a logical sequence in a story”; they could either choose one example to be more coherent or mark both examples as equally (in)coherent. The average human performance on the task is 94.5%,⁷ showing a high agreement between the human annotators and my annotation method. I also measure the inter-annotator agreement, by calculating QWK between the two annotators,⁸ and find it equals to 84.5%, indicating an ‘*almost perfect*’ agreement (Landis and Koch, 1977).

- While a synthetically generated example will most probably be less plausible than its original version, it still could be coherent with respect to the dataset as a whole. I address this by limiting the final evaluation to PRA where I only rank a coherent example against its incoherent version and I do **not** employ TPRA where coherent examples are compared to all the incoherent ones in the dataset.

Table 5.2 shows a few examples from the cloze_swap and cloze_rand datasets. By adding the incoherent examples, the final size of each dataset becomes 195,806 for training, 3,742 for dev and 3,742 for testing. The coherent examples are labeled with a score = 1 and the incoherent ones are given a zero score.

5.2.3 Experiments

Direct evaluation I evaluate the models listed at the beginning of the chapter directly on the two CCD test sets (cloze_swap and cloze_rand); the models are pre-trained on the WSJ as explained in Chapter 4. I also test MTL-GR-based models (MTL_{GRs}, MTL_{GRs}+ELMo, MTL_{GRs}+BERT) pre-trained on Yahoo posts to investigate the effect of pre-training domains. These models achieved, overall, high performance on the WSJ and GCDC datasets and I select the Yahoo domain as it has the highest inter-rater agreement in the GCDC (Table 4.2). In order to evaluate Yahoo models with PRA, I use their variants

⁷One annotator scored 96% and the other 93%.

⁸I note that each pair could be annotated by the judges with one of three possible classes: the first example in the pair is more coherent than the second, it is a tie (i.e., both examples are equally (in)coherent), or the second example is more coherent than the first.

presented in §4.5.2.3 that exploit a final regression layer to predict a real-valued score. I do not employ the multi-class prediction models as it will be difficult to rank the examples if both the coherent and incoherent texts were assigned the same class in evaluation. Throughout this chapter, any use of Yahoo models refers to the ranking models.

Fine-tuned evaluation The test CCD domain is different from the training one (WSJ or Yahoo). For instance, around 52% of the CCD vocabulary is out of the WSJ training documents. In addition, the average sentence length in the CCD is 10, while the average sentence length in the WSJ is 24. Therefore, in order to further investigate domain shift effects, I fine-tune the pre-trained models on each of the `cloze_swap` and `cloze_rand` training sets and re-evaluate performance on the respective test sets. Specifically, each example in the training set is fed to the pre-trained coherence model (pre-trained on WSJ or Yahoo) and the pre-prediction representation of the model (which is further explained at the end of this section) is subsequently fed to an MLP classifier that is fine-tuned to discriminate between coherent and incoherent representations. The MLP classifier is fine-tuned over the representations extracted for the dev set and finally tested on the test sets of the CCD.⁹ Training and testing are done for `cloze_swap` and `cloze_rand` separately.

I use a regression MLP layer with size 100, followed by a sigmoid and minimise the MSE between the gold labels (0 or 1) and the predicted scores. For optimisation, I use Adam (Kingma and Ba, 2015) with L2 regularisation where the penalty rate is tuned on the dev set using the values $\{0.00001, 0.0001, 0.001, 0.01\}$; then, the best value is automatically used for testing (Conneau and Kiela, 2018). I also train the classifier using early stopping, where I stop training if the accuracy (PRA) on the dev set does not increase for 5 epochs, while also setting the maximum number of epochs to 200. I use mini-batches of size 64.

This evaluation setup examines the transferability of the model output representations to a new domain. It is also efficient since I only fine-tune the MLP layer and not the whole coherence model. My aim in this chapter is to create a fast and efficient evaluation framework for neural discourse models as a further examination step after the models are developed and tuned on their respective datasets, and not to provide a new dataset to train and test the models from scratch.

I now describe how I extract the pre-prediction layer from each model. For the MTL and STL models, it is the document vector produced by Eq. 3.6, whereas for the LCD-based models, it is vector O in Eq. 2.9. With regard to the LC model, the pre-prediction representation is the clique embedding resulting from applying a window approach to the two sentences (the output of Eq. 2.5). Finally, for the EGrid approach, I first use the Brown coherence toolkit to represent each input sentence pair as an EGrid matrix.

⁹This is similar to how probing tasks are utilised to investigate different linguistic properties of text (Shi et al., 2016; Adi et al., 2017; Conneau et al., 2018; Ettinger et al., 2018; Hewitt and Manning, 2019; Liu et al., 2019b).

Model	Direct		Fine-tuned	
	cloze_swap	cloze_rand	cloze_swap	cloze_rand
MTL _{GRs}	0.693	0.513	0.888	0.657
MTL _{GRs} +ELMo	0.818	0.521	0.920	0.665
MTL _{GRs} +BERT	0.735	0.533	0.885	0.542
MTL _{POS}	0.692	0.527	0.898	0.633
MTL _{POS} +ELMo	0.803	0.531	0.920	0.641
MTL _{POS} +BERT	0.763	0.523	0.897	0.563
STL	0.742	0.485	0.835	0.537
STL+ELMo	0.786	0.500	0.919	0.647
STL+BERT	0.753	0.525	0.847	0.561
LC	0.707	0.505	0.763	0.513
LCD-L	0.745	0.545	0.884	0.652
LCD-fastText	0.748	0.821	0.948	0.939
LCD-ELMo	0.750	0.680	0.957	0.923
LCD-BERT	0.754	0.710	0.967	0.948
CNN-EGrid _{ext}	0.846	0.534	0.881	0.688
MTL _{GRs} (Yahoo)	0.599	0.498	0.660	0.499
MTL _{GRs} +ELMo (Yahoo)	0.613	0.528	0.783	0.529
MTL _{GRs} +BERT (Yahoo)	0.553	0.504	0.591	0.514

Table 5.3: The PRA values of evaluating the models on the large-scale CCD test sets either directly or with fine-tuning. All the models are pre-trained on the WSJ, except the last three which are pre-trained on Yahoo.

I then input the matrix to the pre-trained CNN-EGrid_{ext} model, and extract the vector resulting from max-pooling the feature maps and feed it to the MLP classifier (see §4.2.2 and Equation 3 in Tien Nguyen and Joty (2017)).

5.2.4 Results

The PRA results of testing the pre-trained coherence models (either directly or with fine-tuning) on the two CCD test sets are displayed in Table 5.3. All the models are pre-trained on the WSJ dataset, except for the last three which are pre-trained on Yahoo posts. As a general observation, we find that, even though cloze_swap and cloze_rand are from the same domain, which is different than the training one, the difference in performance on the two test sets is substantial. This indicates that the models are capable of capturing syntactic corruptions more efficiently than semantic ones, even with no fine-tuning (with the exception of LCD-fastText). We also notice that fine-tuning boosts the discriminative power of the models. I further analyse these results from two perspectives: the model architecture and pre-training domain.

Model architecture Looking at the performance of RNN-based models (the first 11 in Table 5.3) on cloze_swap with direct evaluation, we find that the models are fairly

able to distinguish the intact examples from the noisy ones, showing their robustness to syntactic corruptions. The sequence-based architecture of the models and its sensitivity to order allow them to pick up on the signal given by misordered syntactic constructions, further supporting the evidence from literature regarding the ability of RNNs to capture syntax (Linzen et al., 2016; Kuncoro et al., 2018; Gulordava et al., 2018; Wilcox et al., 2019). The `cloze_swap` test data is different from the training WSJ one, which means that the models are able to capture surface syntactic patterns with no recourse to semantic or lexical cues, enabling them to generalise to new domains. This observation is similar to the one obtained by Gulordava et al. (2018) who found that an RNN language model is able to capture long-distance number agreement even in nonsensical sentences that are grammatical but not meaningful.

Nonetheless, all the RNN models fail to detect semantic alterations when directly evaluated on `cloze_rand`, giving a performance close to random (50%), despite the fact that `cloze_rand` is from the same domain as `cloze_swap`. Although fine-tuning generally improves performance on `cloze_rand`, performance remains poor relative to the one obtained on `cloze_swap` and even stays close to chance with some models (e.g., STL and LC). The maximum PRA yielded by RNN-based models with fine-tuning on `cloze_rand` is 0.665 by `MTLGRs+ELMo`. Semantic corruptions are indeed more challenging to detect and require deeper understanding of natural language. It is, therefore, expected that the models will attain accuracies lower than the ones yielded on `cloze_swap`. However, obtaining accuracies close to random with direct evaluation reveals the models’ inability to identify semantic relations, even when enhanced with contextualised embeddings that are capable of capturing semantic features.

On the other hand, non-sequential models that build sentence representations by averaging their word vectors (LCD-fastText, LCD-ELMo and LCD-BERT) perform on par with their RNN counterpart (LCD-L) on `cloze_swap`, using direct evaluation, yet there is a substantial improvement of their performance on `cloze_rand` over all the other models. What is more surprising is the superior performance of LCD-fastText in detecting semantic perturbations (PRA = 0.821). This suggests that a simple averaging method can be more powerful than sophisticated sequential models in representing sentence meaning. Furthermore, the linear transformations applied by LCD models to sentence pairs help increase the expressive power of the models and facilitate the learning of richer representations. Additionally, LCD-fastText, LCD-ELMo and LCD-BERT are significantly boosted with fine-tuning on the two test sets. The highest PRA is achieved by LCD-BERT on `cloze_swap` and `cloze_rand` (0.967 and 0.948 respectively), with the latter being on par with human performance (0.945), further motivating the transferability of the approach.

The comparison between sequential and non-sequential models can be better illustrated by looking at the performance of LCD-ELMo or LCD-BERT on `cloze_rand` (direct or

fine-tuned) vs. the RNN-based methods that also use ELMo or BERT. The difference in the performance of the two approaches suggests that it is not only contextualized embeddings that impact performance, but also the architectural decision of how to build higher representations from these embeddings. This also agrees with the findings of Conneau et al. (2017) who illustrated how encoding methods impact model transferability to other tasks, even if the encoders were pre-trained on the same data.

Interestingly, we find that the best model on cloze_swap with direct evaluation is CNN-EGrid_{ext}, with 0.846 accuracy. This could be explained by the fact that the used EGrids do not account for pronouns, while pronouns are widely used in the dataset to tie the sentences (§5.2.1). On closer inspection, I find that 67% of the coherent grids do not contain a subject entity in the second sentence as the subject is usually a pronoun, which gives the model a straightforward signal to discriminate between coherent and incoherent examples. However, the performance of CNN-EGrid_{ext} on cloze_rand is close to random, similar to the RNN-based models.

It is also obvious that the models are further enhanced by integrating contextualised embeddings as they help mitigate the out-of-domain problem; for instance, the second highest PRA on cloze_swap direct evaluation is 0.818 by MTL_{GRs}+ELMo, whereas its basic version (MTL_{GRs}) scores 0.693. Finally, regarding the integration of syntactic features in training (MTL_{GRs} and MTL_{POS}), we find that with direct evaluation they surprisingly do worse than STL on cloze_swap. However, with further fine-tuning, they surpass the STL model on both cloze_swap and cloze_rand. This suggests that the latent representations of the syntactic models carry useful discourse-relevant information that enhances the discriminative power of the MLP classifier.

Pre-training domain I now move to examining the effect of the pre-training domain by comparing the same models when trained on synthetic WSJ and the realistic Yahoo domain. The results of pre-training with Yahoo are displayed in the last three rows of Table 5.3. Regarding performance on cloze_swap with direct evaluation, we find that when the models are pre-trained on Yahoo they perform substantially worse than when pre-trained on the WSJ.¹⁰ With fine-tuning, WSJ models remain substantially better than their Yahoo counterparts on cloze_swap. This shows that the pre-training domain impacts the ability of models to capture linguistic information. I conjecture that the nature of the synthetic WSJ training data with its syntactic regularities that are corrupted in the incoherent documents lends itself to the models built to capture sequential information. Such models are able to memorise these patterns and thereby discriminate between an intact text and an unordered one. In contrast, in the realistic domain, the difference

¹⁰As a further investigation into the realistic domain, I directly evaluate MTL_{GRs}+ELMo pre-trained on Enron emails. The model obtains 0.616 on cloze_swap and 0.505 on cloze_rand, which is very close to the Yahoo results.

between the syntactic structures of coherent documents vs. less coherent ones is less pronounced, giving a weaker signal to the models and thus leading to a lower performance (as discussed earlier in §4.5.1). Finally, as for `cloze_rand`, as expected, pre-training on Yahoo also performs close to chance, even when the models are fine-tuned.

5.3 Controlled Linguistic Alterations Dataset

In the previous section, I created a large-scale out-of-domain corpus to investigate what coherence approaches learn. I compiled two datasets, the first exhibits broken syntactic patterns and the second contains semantic/rhetorical corruptions. As further examination, in this section, I carry out a systematic error analysis and manually craft a dataset, from a domain similar to the WSJ, of controlled linguistic alterations (CLAD) and test the sensitivity of the models to these alterations. The idea is to create a test set with a few coherent well-written examples, apply minor perturbations to them and observe the changes in the predictions of the different models. This dataset applies black-box adversarial evaluation (§2.6) to test the robustness of the models to adversarial examples, where we do not have access to model parameters. It is also inspired by the work of Zhu et al. (2018) who investigated the effect of applying systematic syntactic and semantic changes to sentences on the degree of similarity between their respective sentence embeddings, in order to test the ability of different sentence encoders to encapsulate semantic and syntactic features.

5.3.1 Dataset

The CLAD includes 30 examples, consisting of two sentences each. Specifically, I extract the sentence pairs from business and financial articles published in the BBC, the Independent and Financial Times in the year 2019, resulting in examples close to the WSJ domain. The selection of the pairs that form the original coherent examples is achieved in a controlled way for the purpose of my evaluation. More concretely, I choose sentence pairs where the subject of the first sentence is pronominalized in the second, and the second sentence begins with this pronoun. This way, the style of the examples is constrained to be similar to a frequently occurring pattern in the WSJ (“S → NP-SBJ VP | NP-SBJ → PRP”) as described earlier in §5.1. I also select the examples so that they are self-contained and do not reference an outer context. The average sentence length of the examples is 24.2 which is close the training WSJ domain. I note that there will still be some variation from the WSJ, especially that the selected articles are much newer and are likely to have different entities. However, I conjecture that this deviation will still not have much impact on my evaluation, which is further elaborated in the results I present later. In addition,

Original	A government paper on Monday found UK and EU firms would be faced with a “a significant new and ongoing administrative burden” in the event of a no-deal Brexit. It found large firms importing and exporting at scale would need to fill in form taking one hour 45 minutes on average and cost £28 per form for each load imported.
Swap	It found large firms importing and exporting at scale would need to fill in forms taking one hour 45 minutes on average and cost £28 per form for each load imported. A government paper on Monday found UK and EU firms would be faced with a “a significant new and ongoing administrative burden” in the event of a no-deal Brexit.
Prefix Insertion	<i>More specifically</i> , it found large firms importing and exporting at scale would need to fill in forms taking one hour 45 minutes on average and cost £28 per form for each load imported. A government paper on Monday found UK and EU firms would be faced with a “a significant new and ongoing administrative burden” in the event of a no-deal Brexit.
Lexical Substitution	<i>The paper</i> found large firms importing and exporting at scale would need to fill in forms taking one hour 45 minutes on average and cost £28 per form for each load imported. A government paper on Monday found UK and EU firms would be faced with a “a significant new and ongoing administrative burden” in the event of a no-deal Brexit.
Random	1- A government paper on Monday found UK and EU firms would be faced with a “a significant new and ongoing administrative burden” in the event of a no-deal Brexit. She spent over a decade at Swiss investment bank UBS before joining the UK Treasury’s council of economic advisers in 1999. 2- Lady Vadera was born in Uganda and moved to the UK as a teenager. It found large firms importing and exporting at scale would need to fill in forms taking one hour 45 minutes on average and cost £28 per form for each load imported.
Lexical Perturbations	A government paper on Monday found UK and EU firms would be faced with a “a significant new and ongoing administrative burden” in the event of a no-deal Brexit. It found large firms importing and exporting at scale would need to fill in <i>cups</i> taking one hour 45 minutes on average and cost £28 per <i>cup</i> for each load imported.
Corrupt Pronoun	A government paper on Monday found UK and EU firms would be faced with a “a significant new and ongoing administrative burden” in the event of a no-deal Brexit. <i>He</i> found large firms importing and exporting at scale would need to fill in forms taking one hour 45 minutes on average and cost £28 per form for each load imported.

Table 5.4: An example from CLAD showing the different methods of creating incoherent examples from the original one in the first row. For ‘Random’ I create two instances: 1- the first sentence is unchanged while the second is randomly selected from other examples, and 2- the first sentence is randomly selected while the second is kept intact.

this makes the test set more expandable, giving future researchers the flexibility to add more examples from the large financial domain.

In order to create the incoherent texts, I apply a set of linguistic changes to the 30 original examples, and thereby each change results in a small test set of coherent and incoherent pairs. The changes are classified as syntactic or semantic and are carefully chosen in order to systematically examine model performance. The following describes how the incoherent instances are constructed and Table 5.4 shows an example; for more examples see Appendix B.

Syntactic datasets The syntactic changes aim at corrupting sentence order while preserving the meaning each sentence conveys. I apply three changes:

- **Swap.** I simply swap the two sentences in each example, the same way I created cloze_swap.
- **Prefix Insertion.** I also swap the two sentences but insert a prefix phrase in the second one. I analyze the WSJ training data and find that the average number of times the first sentence in a document starts with a pronoun is 0.02 (and never with ‘he’ or ‘she’) which is significantly less than the average number of times a sentence starts with a pronoun (regardless of its position) which is 0.07. This difference is not maintained in the shuffled documents so might give a signal to the models to

detect that a swapped pair that starts with a pronoun is less coherent. In order to examine if such positional information plays a role in prediction, I insert a phrase, before the subject pronoun after swapping the sentences, that does not change the propositional content (e.g., ‘More specifically’, ‘However’, etc.). I observe whether this insertion will change the prediction of the model.

- **Lexical Substitution.** I swap the two sentences but replace the subject pronoun in the second one with ‘the + a general noun’ that substitutes the subject in the first sentence (e.g., ‘the company’, ‘the woman’, etc.). This setup should also test whether models rely on pronouns as indicators for sentence order, especially that pronominalizing subjects in subsequent utterances is a common pattern in the training data.

Semantic datasets In contrast to the syntactic datasets, the semantic ones aim at maintaining the main syntactic patterns but perform semantic alterations that would result in a nonsensical discourse.

- **Random.** Similar to `cloze_rand`, I keep the first sentence intact and select a second random sentence from the dataset. I constrain the selection so that the subject pronoun is different from the subject pronoun in the original sentence.¹¹ I create another random pair with the same constraint yet this time I replace the first sentence and keep the second intact. Thus each original example will be compared with two examples alleviating the impact that the language model probabilities of sentences might have on the final score.
- **Lexical Perturbation.** I investigate the robustness of the models to minor lexical changes that result in incoherent meaning, by replacing one word in either of the two sentences (if the word is repeated, I change that too). I choose a replacement word from the training vocabulary of the WSJ with the same part-of-speech tag.
- **Corrupt Pronoun.** I replace the subject pronoun in the second sentence with another pronoun that cannot reference anything in the first sentence. I also replace any other relevant pronouns in the second sentence to make it grammatical (e.g., replacing the possessive pronouns referring to the subject pronoun to reflect the change). With this method, I test whether the models are capable of resolving coreferences or just rely on syntactic patterns.

¹¹I also take into account that some subjects could be referred to by ‘he’, ‘she’ or ‘they’ and thus factor that into the selection.

Dataset	Swap	Prefix Insertion	Lexical Substitution	Random	Lexical Perturbations	Corrupt Pronoun	All	TPRA
# Comparisons	30	30	30	60	30	30	210	6,300
MTL _{GRs}	0.900	0.833	0.833	0.566	0.566	0.700	0.709	0.699
MTL _{GRs} +ELMo	0.966	0.933	0.966	0.466	0.500	0.666	0.709	0.682
MTL _{GRs} +BERT	0.933	0.966	0.933	0.450	0.466	0.533	0.676	0.713
MTL _{POS}	0.833	0.766	0.800	0.483	0.633	0.500	0.642	0.642
MTL _{POS} +ELMo	0.933	0.866	0.866	0.483	0.433	0.500	0.652	0.666
MTL _{POS} +BERT	0.833	0.900	0.866	0.500	0.533	0.466	0.657	0.682
STL	0.833	0.766	0.800	0.500	0.466	0.633	0.642	0.618
STL-ELMo	0.900	0.933	0.900	0.566	0.566	0.533	0.709	0.701
STL-BERT	0.900	0.900	0.900	0.516	0.633	0.633	0.714	0.716
LC	0.800	0.766	0.866	0.516	0.500	0.533	0.642	0.660
LCD-L	0.933	0.866	0.833	0.616	0.533	0.600	0.714	0.691
LCD-fastText	0.800	0.800	0.766	0.783	0.733	0.800	0.780	0.676
LCD-ELMo	0.933	0.933	0.800	0.833	0.666	0.666	0.809	0.777
LCD-BERT	0.866	0.933	0.866	0.783	0.800	0.766	0.828	0.722
CNN-EGrid _{ext}	0.833	0.800	0.766	0.716	0.533	0.566	0.704	0.658
MTL _{GRs} (Yahoo)	0.500	0.433	0.500	0.500	0.500	0.433	0.480	0.486
MTL _{GRs} +ELMo (Yahoo)	0.533	0.533	0.566	0.450	0.533	0.500	0.509	0.497
MTL _{GRs} +BERT (Yahoo)	0.466	0.500	0.500	0.483	0.533	0.633	0.514	0.524

Table 5.5: Results on the CLAD for the different models. All models are trained on the WSJ, except for the last three rows trained on Yahoo. # comparisons refers to the number of measured pairwise rankings between coherent and incoherent documents. “All” is the ranking accuracy when comparing each coherent example against *its* incoherent versions across all the datasets. “TPRA” refers to the ranking accuracy when comparing the coherent examples against *all* the incoherent ones in the whole dataset, not just their noisy counterparts.

5.3.2 Results

Similar to the direct evaluation presented in §5.2.3, I test the coherence models on the CLAD and report the results in Table 5.5. The table displays the PRA of each syntactic or semantic dataset, by comparing each coherent document with its noisy version in this particular dataset. I also report the “All” results of ranking the coherent examples against *their* incoherent versions but across all the datasets (i.e., the original example in Table 5.4 will be compared with the rest of the examples in the table, and that will be done for each coherent text in the dataset). Additionally, I perform comparisons across different examples by calculating TPRA which ranks the coherent examples against all the incoherent ones in the whole dataset, not just their noisy versions.

Due to the small size of the dataset, I analyse the main noticeable trends and not the subtle differences in performance. Overall, the results on the CLAD agrees with the CCD results. We find that all the WSJ models are able to detect syntactic broken patterns; they manage to capture the swap examples even in the cases where a prefix is inserted or the subject pronoun is substituted with a lexical item in the reversed pair. The performance on Prefix Insertion and Lexical Substitution further indicates that the models are capable of capturing the relevant syntactic patterns and do not rely solely on pronouns or positional features. However, performance drops when evaluated against the

semantic examples which suggests that the models fail to detect topical or rhetorical shifts and unresolved references. More specifically, on the Random dataset, RNN-based models perform close to chance, despite the fact that the examples are close to the WSJ domain. For instance, $MTL_{GRs}+ELMo$ is the strongest model on the syntactic datasets, close to 100%, but yields 0.466 on the Random dataset. The same model achieves the second best performance on `cloze_swap` with direct evaluation (0.818) which increases to 0.920 with fine-tuning, in addition to its high performance on the WSJ (0.960 in Table 4.5). All these results suggest that the model might be memorising relevant syntactic patterns and therefore is able to recognise when these patterns are broken, while paying less attention to the underlying meaning. The architecture of the model and the nature of its training domain motivates such behaviour.

On the other hand, the averaging models (LCD-fastText, LCD-ELMo and LCD-BERT) are the only models with a relatively satisfying performance on the Random dataset. This is further validated when comparing them to their RNN counterpart (LCD-L) or other ELMo or BERT based models, further corroborating the role of sentence composition methods. We also observe that the $CNN-EGrid_{ext}$ model surpasses other RNN-based models on the random dataset. The reason could be that the entities in the two sentences of an example are different, resulting in a sparser grid; for instance in the original example in Table 5.4, ‘firms’ is mentioned in the two sentences, while in the two Random examples, it is only mentioned in one.

Regarding the performance on Lexical Perturbations, the models are not sensitive to minor lexical changes even if they result in implausible meaning. This outcome is expected due to the difficulty of the task as it requires a deeper understanding of meaning. LCD-BERT followed by LCD-fastText, however, better capture these lexical intricacies and provide more promising performance. Furthermore, these two models also outperform other more complex approaches on the corrupt pronoun pairs further demonstrating some ability to resolve pronominal reference. Nonetheless, with a more comprehensive evaluation (TPRA), where all the coherent examples are ranked against the incoherent ones in the whole dataset, the performance of LCD-fastText drops unlike the one yielded when the comparison is done between versions of the same example (All = 0.780 vs. TPRA = 0.676). The LCD ELMo and BERT models are better at generalisation with the highest TPRA (0.777) yielded by LCD-ELMo, which further indicates the power of contextualised embeddings.

I finally shed light on the effect of the pre-training domain. We can see that the models pre-trained on Yahoo fail on all the datasets, providing a close to random performance. This is expected since the test examples are from a domain different from Yahoo posts. Furthermore, realistic data is different from synthetic one; in the latter, there are prominent syntactic regularities in the coherent articles that are corrupted in the shuffled documents,

yet in the realistic domain such syntactic features are less pronounced. This makes it more challenging for Yahoo models to detect the swapped examples.

5.4 Summary

I have presented an evaluation framework for discourse coherence models that consists of two main datasets of sentence pairs. The first is a large-scale dataset, from a short stories domain, that exhibits syntactic or semantic corruptions on which I directly evaluate the pre-trained coherence models, or allow further fine-tuning using their pre-prediction layer to adapt to the new test domain. The evaluation on this dataset reveals that RNN-based coherence models memorise syntactic patterns that co-occur in coherent texts yet fall short in capturing semantic aspects that play a key role in discourse, even though both the syntactic and semantic test examples are from the same domain. With further fine-tuning, the models are substantially boosted, yet the gap in performance on the syntactic corruptions and the semantic ones still holds. Furthermore, adding contextualised embeddings to models improves their performance. On the other hand, semantic relations are better captured by approaches that encode sentences by averaging their word representations, then apply a suite of linear transformations over sentence pairs to increase the expressive power of the models (the LCD models). These models are, overall, more transferable than my hierarchical approach. With regard to the effect of the pre-training domain, when the models are pre-trained on the WSJ they perform consistently better than when pre-trained on Yahoo posts, with or without fine-tuning. This could be attributed to the nature of these domains; the WSJ is a synthetic dataset where coherent documents exhibit syntactic regularities that are corrupted in their incoherent versions, giving a straightforward signal to the models built to capture sequential information.

The second dataset I create is a small-scale one of controlled linguistic alterations to systematically examine the susceptibility of the models to changes in syntax or semantics. I select the examples for this dataset from the financial domain, close to the WSJ one. I evaluate different coherence models on this test data and reach the same conclusion as in the case of the large-scale dataset: RNN-based models are able to detect broken syntactic patterns but fail to model semantic or rhetorical features. Their ability to capture syntax still remains when subject pronouns are substituted with general nouns or a prefix phrase is inserted, indicating that they do not rely on positional features. These models are also not sensitive to minor lexical perturbations nor can resolve pronouns. On the other hand, the LCD models that average word vectors are again more capable of capturing semantic relations. As for evaluation after pre-training on Yahoo, we find that the models fail at both syntactic or semantic tasks, which is expected as the test examples are close to the WSJ domain but different from the Yahoo one.

In this chapter, I aim to provide an evaluation setup for discourse models that researchers could leverage to test and better understand their models. Furthermore, my dataset of controlled linguistic alterations provides a framework that could be further extended to include more examples following my same constraints or create new constraints to examine other aspects of discourse coherence.

APPLICATION OF COHERENCE MODELS

In this chapter, I investigate applications for discourse coherence models; I specifically focus on the learner domain and examine the efficacy of integrating my coherence models to neural Automated Essay Scoring (AES) systems. I demonstrate that state-of-the-art approaches to AES are not well-suited to capturing adversarially crafted input of grammatical but incoherent sequences of sentences. Therefore, I propose a framework for combining and jointly training a discourse model with a state-of-the-art neural AES system in order to enhance its ability to capture connectedness features between sentences. I first evaluate the integration of the local coherence (LC) model (§2.4.1 and §4.2.1) in the joint framework, with different parameter sharing setups, and experimentally examine its effectiveness on both the AES task and the task of flagging adversarial input. I then assess the impact of leveraging MTL-based approaches by incorporating MTL_{GRs} , $MTL_{GRs}+ELMo$ or $MTL_{GRs}+BERT$ as the discourse component in the framework. The experiments show that my joint learning (JL) framework can efficiently detect adversarial input while maintaining a high performance in predicting a holistic essay score.

The chapter is structured as follows. In §6.1, I discuss previous AES models and focus on the systems that model discourse coherence. In §6.2, I explain how adversarial evaluation has been used to validate AES systems. Next in §6.3, I present the AES dataset I utilise for my experiments and detail the evaluation metrics I leverage. In §6.4, §6.5 and §6.6, I explain the used AES systems, coherence models and joint learning framework that combines both respectively. After that, in §6.7, I detail my experimental setup and then present my results and analysis in §6.8. Finally, in §6.9, I conclude and summarise the chapter. I note that this chapter is based on a long paper published in the 16th Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2018) (Farag et al., 2018).

6.1 Approaches to automated essay scoring

In this section, I give a brief overview about existing AES approaches (traditional or neural) with special focus on the approaches that model discourse coherence.

6.1.1 Traditional approaches

The task of AES focuses on automatically analysing the quality of writing and assigning a holistic score to essays. Traditionally, AES models have exploited a wide range of manually-tuned linguistic features that are associated with writing competence (Burstein et al., 2003; Rudner et al., 2006; Yannakoudakis et al., 2011; Shermis and Hammer, 2012; Williamson et al., 2012; Andersen et al., 2013; Chen and He, 2013; Phandi et al., 2015; Zupanc and Bosnić, 2017). Some approaches relied on shallow textual features as “proxies” for writing quality such as length and counts of POS tags (Page, 1968), or word ngrams and number of verbs (Rudner and Liang, 2002). Other systems exploited deeper features such as: properties related to grammar, style, discourse and lexical complexity (Attali and Burstein, 2006), semantic similarity between train and test essays using LSA (Landauer, 2003), or syntactic and lexical features (e.g., phrase structure rules and lexical ngrams) (Yannakoudakis et al., 2011).

Coherence in traditional AES models As discourse coherence is an important dimension in writing quality, a large body of AES research has been devoted to modeling coherence and organisation in student essays. For instance, Centering theory was computationally utilised to assess essays by using rough entity shifts as a proxy for the level of incoherence (Miltisakaki and Kukich, 2000) or leveraging the continuity concept of the theory (Rus and Niraula, 2012b). Other work exploited EGrid representations of essays (Burstein et al., 2010; Palma and Atkinson, 2018). Essays were also assessed by tracking their topic development via *topic chains* that model organisation in writing (Rahimi et al., 2015) or via HMMs (Liu et al., 2013). Furthermore, estimating semantic similarity between different text parts, inspired by LSA, was widely employed to approximate writing coherence (Higgins et al., 2004; Higgins and Burstein, 2007; Yannakoudakis and Briscoe, 2012; Palma and Atkinson, 2018). Other models utilised lexical chains of semantically-related words (Somasundaran et al., 2014) or RST relations (Feng et al., 2014; Huang et al., 2018).

6.1.2 Neural approaches

More recently, advances in deep learning have shown the efficacy of end-to-end neural approaches on the task of AES (Alikaniotis et al., 2016; Taghipour and Ng, 2016; Dong

and Zhang, 2016; Dong et al., 2017; Riordan et al., 2017; Farag et al., 2017; Wang et al., 2018; Zhang and Litman, 2018). Much of this research has focused on the Automated Student Assessment Prize (ASAP) dataset (as will be discussed in §6.3.1) and showed the superiority of neural models over traditional feature-engineered systems in this domain of student essays. For instance, Alikaniotis et al. (2016) developed a deep Bi-LSTM network, augmented with score-specific word embeddings that capture usage information for words. Taghipour and Ng (2016) investigated various recurrent and convolutional architectures and found that an LSTM layer followed by a Mean over Time operation achieves the best results, as will be explained in §6.4. Dong and Zhang (2016) showed that a two-layer CNN outperformed other baselines (e.g., Bayesian Linear Ridge Regression) on both in-domain and domain-adaptation experiments on the ASAP essays. Dong et al. (2017), generated sentence representations using a CNN followed by attention pooling in a prompt-independent fashion, whereas Zhang and Litman (2018) showed that a co-attention mechanism between the prompts and the essays responding to them surpasses the prompt-independent approach.

Coherence in neural AES models Efforts have also been directed to building more discourse-aware neural AES models (Tay et al., 2018; Mesgar and Strube, 2018; Liu et al., 2019a; Mim et al., 2019; Nadeem et al., 2019). For instance, Tay et al. (2018) trained an LSTM network to predict a holistic essay score by generating coherence features between different essay parts. A coherence feature is learned between two hidden vectors at different time steps of the LSTM, separated by a window of a pre-defined size, by computing a bilinear product between the vectors. By doing so, the model learns semantic relationships between snapshots from the essay. The final prediction layer averages the LSTM hidden states and concatenates the output with the generated coherence features to predict the overall score. Mesgar and Strube (2018) generated neural coherence vectors (§2.4.1) and combined them with feature vectors generated by open-source AES models to train a regression model to predict the final essay scores. The work of Nadeem et al. (2019) also integrated learning coherence-related properties as a pre-training task to AES, which will be further discussed in §6.4.

6.2 Evaluation against adversarial input

In §2.6, I discussed that machine learning models could be tricked with adversarial input which motivates the importance of adversarial training and evaluation to these models. Several studies have focused, specifically, on testing the robustness of AES engines against subversions by writers who may exploit their knowledge about the workings of the systems to maximize their scores (Powers et al., 2002; Yannakoudakis et al., 2011; Lochbaum

et al., 2013; Bejar et al., 2014; Higgins and Heilman, 2014; Zhang et al., 2016; Yoon et al., 2018). This testing is imperative to ensure the validity of a system before deployment, especially in high-stakes assessment. For instance, Powers et al. (2002) asked writing experts to write essays to trick the e-rater system (v1) (Burstein et al., 1998), after briefing them on how the system works.¹ The participants managed to fool the system into assigning higher-than-deserved grades, most notably by simply repeating a few well-written paragraphs several times. Yannakoudakis et al. (2011) and Yannakoudakis and Briscoe (2012) created and used an adversarial dataset of well-written texts and their random sentence permutations, which they released in the public domain, together with the grades assigned by a human expert to each piece of text. Yannakoudakis and Briscoe (2012) succeeded in capturing such adversarial input by measuring semantic similarity between sentences; however, their test data is quite small, consisting of 12 scripts in total. Higgins and Heilman (2014) demonstrated the susceptibility of AES systems to gaming strategies including padding an essay with: multiple copies of the same essay, random words from the prompt or random academic words. Furthermore, there have been attempts to empirically address adversarial inputs by incorporating pre-assessment techniques that flag texts that could potentially fool the system. Some of these techniques focused on capturing shallow features such as essay length (too long or too short), repetitions of words and sentences or repeating the prompt (Attali and Burstein, 2006; Zhang et al., 2016; Yoon et al., 2018). Other approaches addressed the detection of off-topic responses (Higgins et al., 2006; Louis and Higgins, 2010; Persing and Ng, 2014; Rei and Cummins, 2016; Li et al., 2017). More recently and subsequent to my work here, Kumar et al. (2020b) created a framework for generating adversarial input by applying a wide variety of syntactic and semantic changes to ASAP essays to lower their quality, and demonstrated the inadequacy of state-of-the-art neural AES approaches to capture such adversaries. They showed that adversarial detection could be *marginally* improved by adversarial training; however, they did not test the effect of adversarial training on the main task of predicting holistic essay scores.

I extend this line of work and examine the robustness of state-of-the-art neural AES models to adversarially crafted input and specifically focus on input related to local coherence; that is, grammatical but incoherent sequences of sentences. I demonstrate that neural models are vulnerable to such input and thereby propose an approach to enhance their ability to detect adversarial incoherent texts.

¹E-rater relies on a wide variety of features associated with structure (e.g., syntactic features), organisation (e.g., discourse features) and content (e.g., vocabulary related features).

Prompt	Size of ASAP Original Dataset	Vocab Size	Avg # Sents	Avg Sent Length	Score Range	Synthetic (Adversarial) Dataset			Combined Dataset Size
						Threshold	#Selected High-Scoring Essays	Total Size	
1	1,783	16,312	22	18	2 – 12	10	472	5,192	3,671
2	1,800	15,074	20	21	1 – 6	5	82	902	2,128
3	1,726	6,658	6	19	0 – 3	3	407	4,477	3,354
4	1,772	5,078	4	22	0 – 3	3	244	2,684	2,748
5	1,805	5,273	6	21	0 – 4	4	258	2,838	2,837
6	1,800	5,737	7	22	0 – 4	4	367	4,037	3,268
7	1,569	10,625	12	16	0 – 30	23	179	1,969	2,285
8	723	12,660	34	20	0 – 60	45	72	792	1,011

Table 6.1: Statistics for each dataset per prompt. For the synthetic dataset, the high scoring ASAP essays are selected based on the indicated score threshold (inclusive) and “Total Size” refers to the number of the ASAP essays selected + their 10 permutations. The combined dataset refers to the original dataset + 4 permutations from the synthetic data for each high scoring essay. ‘Vocab Size’ refers to the number of unique words.

6.3 Dataset and evaluation

6.3.1 Dataset

Before I elucidate my joint learning approach, I describe, in this section, the dataset used in my experiments and explain the creation of adversarial examples. I use the Automated Student Assessment Prize (ASAP) dataset created in 2012 for an AES competition by Kaggle and sponsored by the Hewlett Foundation.² The dataset contains 12,976 essays written by students ranging from Grade 7 to Grade 10 in response to 8 different prompts. I follow the ASAP data split by Taghipour and Ng (2016),³ and apply 5-fold cross validation in all the experiments using their same train/dev/test splits, where in each fold, 60% of the essays are used for training, 20% for development and 20% for testing. For each prompt, the test predictions across the 5 folds are aggregated and evaluated together. I refer to this dataset as *original* ASAP dataset.

In order to create adversarial input, for each prompt, I select high-scoring original essays (based on a pre-defined score threshold, Table 6.1) that are assumed coherent, and create 10 permutations per essay by randomly shuffling its sentences. The reason why I limit my selection in adversarial creation to high-scoring essays is two-fold. First, my aim is to evaluate the models against essays written in “bad faith”, that is, sets of well-formed sentences that have been rote-learned and re-produced in an attempt to maximise the assigned score in the knowledge that an automated system is not checking for coherence. Second, poorly-written essays could be highly incoherent and therefore it is unreliable to teach the model to rank them higher than adversarial texts created by permuting well-written essays. I refer to the set of essays that contains the adversarial examples along with their original counterparts as *adversarial* or *synthetic* dataset. I note that in the synthetic data I keep the same train/dev/test splits as the original one and coherent

²<https://www.kaggle.com/c/asap-aes>

³<https://github.com/nusnlp/nea/tree/master/data>

essays are assigned a score of one while incoherent essays are given a zero score.

Finally, in order for the joint learning setup to learn from both original and adversarial essays, I augment the original ASAP dataset with a subset of the synthetic essays, and refer to the resulting dataset as *combined* dataset. Specifically, I randomly select 4 permutations per essay to include in the combined set.⁴ In this dataset, adversarial essays are annotated with the lowest possible score in the score range of their respective prompts (zero in most prompts).

More concisely, I end up with 3 versions of the dataset: ASAP original, synthetic and combined; the statistics for the 3 datasets are detailed in Table 6.1.⁵ The combined data is only leveraged for training but, at test time, I only evaluate on original and synthetic test sets. I do not evaluate a combined test set as I believe that this requires human annotations for the adversarial essays, because an incoherent essay with well-formed sentences may be graded higher than a poorly-written essay that is kept intact without shuffling. This assumption was further proven by the expert grading of the outlier texts compiled by Yannakoudakis et al. (2011). However, it is guaranteed that those adversarial essays are of lower quality than their original counterparts and hence I restrict adversarial evaluation to comparisons between synthetic essays and their original versions.

6.3.2 Evaluation

I test the performance on the original ASAP dataset using Quadratic Weighted Kappa (QWK). I focus on QWK as it is the official evaluation metric for the ASAP competition,⁶ but also report Pearson’s and Spearman’s correlation coefficients in Appendix C. I report the QWK score for each prompt and also the average QWK across all prompts after applying Fisher transformation (§2.7) to show the overall performance of each model as recommended by the ASAP competition. As for synthetic data evaluation, similar to previous chapters, I utilise PRA and TPRa to rank the essays in the test data, I report both values per prompt as well as their average value across prompts. For more details about the used evaluation metrics, see §2.7. I note that, during training, scores are mapped to a range between 0 and 1 (similarly to Taghipour and Ng (2016)), and then scaled back to their original range during evaluation.

⁴This is primarily done to keep the data balanced: initial experiments showed that training with all 10 permutations per essay harms AES performance, but has negligible effect on adversarial input detection.

⁵My combined and synthetic datasets are available at https://github.com/Youmna-H/Coherence_AES

⁶<https://www.kaggle.com/c/asap-aes/overview/evaluation>

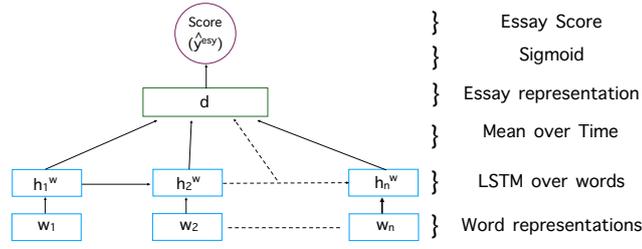


Figure 6.1: The LSTM AES model by Taghipour and Ng (2016) ($LSTM_{T\&N}$).

6.4 Neural AES models

In this section, I describe the AES models I use in my work; all the models learn to predict a holistic essay score that indicates its quality. AES models are typically trained and tested on original data.

$LSTM_{T\&N}$ Taghipour and Ng (2016) trained and evaluated an LSTM model on the ASAP original dataset; I refer to their model as $LSTM_{T\&N}$. The model performs the task of predicting a holistic score for an essay from its sequence of words as can be seen in Fig. 6.1. Each word in an input essay is initialised from a pre-trained embedding space and then a one-layer LSTM⁷ is exploited to encode the sequence of word representations (w_t). Subsequently, a *Mean over Time* operation is applied to simply average the hidden representations ($h_t^w \in \mathbb{R}^{dim^w}$, where dim^w is a hyperparameter) of the LSTM⁸ to produce an essay vector (d) that is scored by a linear transformation followed by a sigmoid function.⁹ The following equations explain how the model scores input essays:

$$\begin{aligned}
 h_t^w &= LSTM(w_t, h_{t-1}^w) \\
 d &= \frac{1}{n} \sum_{t=1}^n h_t^w \\
 \hat{y}^{esy} &= \sigma(W^d \cdot d)
 \end{aligned} \tag{6.1}$$

where $W^d \in \mathbb{R}^{dim^w}$ is a learned weight and n is the number of words in the essay. The network optimises the MSE loss between the predicted scores and the ground-truth ones:

$$L_{esy} = \frac{1}{N} \sum_{i=1}^N (y_i^{esy} - \hat{y}_i^{esy})^2 \tag{6.2}$$

⁷Unidirectional LSTM outperformed Bi-LSTM (Taghipour and Ng, 2016).

⁸For the detailed LSTM equations, see Eq. 2.2.

⁹I note that the authors achieved a bit higher results when averaging ensemble results of their LSTM model together with CNN models. I use their main LSTM model which, for the purposes of my experiments, does not affect my conclusions.

where N is the number of training essays. We can see from the above description and from Fig. 6.1 that $\text{LSTM}_{\text{T\&N}}$ ignores sentence boundaries and thus might be agnostic to discourse-related features, as will be verified in §6.8.

LSTM_{T&N}+ELMo I extend $\text{LSTM}_{\text{T\&N}}$ and create another version of the model where I initialise the input words with ELMo vectors. Following my approach in §3.1.1, I only take the top layer in the triple-layer ELMo representation.

LSTM_{T&N}+BERT Similarly, I create a version of $\text{LSTM}_{\text{T\&N}}$ by leveraging BERT embeddings, following §3.1.1.

LSTM_{T&N}-comb The three aforementioned $\text{LSTM}_{\text{T\&N}}$ -based models are trained on the original ASAP essays, following Taghipour and Ng (2016). Nevertheless, I train a version of $\text{LSTM}_{\text{T\&N}}$ on the combined dataset (i.e., on both original and synthetic essays). I use this approach in order to test the performance when the model sees adversarial examples during training.

NLI-DM-BCA The $\text{LSTM}_{\text{T\&N}}$ model is designed to encode the input essay as a sequence of words, with no explicit modeling for discourse features or sentence interactions. It would be interesting to test a more discourse-aware state-of-the-art AES system, and therefore, I evaluate the model by Nadeem et al. (2019). They adopted a hierarchical LSTM with bidirectional context and attention (BCA) (Nadeem and Ostendorf, 2018). BCA employs an LSTM to generate sentence representations from words then a second LSTM over the sentence vectors to construct a document representation. The first LSTM generates word hidden representations (h_{it}^w , for the t^{th} word in the i^{th} sentence), in addition to a “look-back” and “look-ahead” context vectors conditioned on preceding and subsequent sentences respectively. The “look-back” vector is constructed by using context attention over the preceding sentence:

$$\begin{aligned}\alpha_{(i-1)t}(w_{it}) &= \frac{\exp(h_{it}^w W^\alpha h_{(i-1)t}^w)}{\sum_{t'} \exp(h_{it}^w W^\alpha h_{(i-1)t'}^w)} \\ c_{(i-1)}(w_{it}) &= \sum_{t'} \alpha_{(i-1)t'}(w_{it}) h_{(i-1)t'}^w\end{aligned}\tag{6.3}$$

where W^α is a bilinear weight matrix. Similarly, the “look-ahead” vector is calculated by using context attention over the subsequent sentence.

$$\begin{aligned}\alpha_{(i+1)t}(w_{it}) &= \frac{\exp(h_{it}^w W^\alpha h_{(i+1)t}^w)}{\sum_{t'} \exp(h_{it}^w W^\alpha h_{(i+1)t'}^w)} \\ c_{(i+1)}(w_{it}) &= \sum_{t'} \alpha_{(i+1)t'}(w_{it}) h_{(i+1)t'}^w\end{aligned}\tag{6.4}$$

The final word representation is the concatenation of the context vectors and the LSTM output: $[c_{(i-1)}(w_{it}), h_{it}^w, c_{(i+1)}(w_{it})]$. Word representations are then aggregated with attention to compose sentence representations and, in turn, sentence vectors are processed with an LSTM followed by attention to generate a document vector (similar to my method in §3.1.2 and §3.1.4). Furthermore, Nadeem et al. (2019) incorporated two pre-training auxiliary tasks to their BCA model. The first task involves predicting Natural Language Inference (NLI) labels for adjacent sentence pairs, and the second, which is coherence-related, predicts the category of discourse markers (DM) connecting each sentence pair (categories include justification, opposition and time relation). The sentence pairs for the DM task were extracted from books from www.smashwords.com, where the second sentence in each pair starts with a discourse marker. They used 87 markers that are mapped to 7 categories. Examples of markers with their categories are: ‘nonetheless’ and ‘however’ for opposition, ‘for example’ and ‘in other words’ for justification and ‘meanwhile’ and ‘simultaneously’ for time relations.

Finally, after pre-training the BCA model on the two tasks, the model is trained to perform essay scoring while fixing the pre-training task-specific word-level attention weights; this model is referred to as NLI-DM-BCA. The resulting AES system aims at capturing discourse features of text by attending to previous and next contexts as well as leveraging the DM pre-training task. The NLI-DM-BCA was trained and evaluated only on prompts 1 and 2 of the ASAP dataset as they are persuasive essays that could benefit from discourse modeling, in addition to TOEFL essays (Blanchard et al., 2013).

6.5 Coherence models

I now present the coherence models I leverage and plug later in my joint learning framework. Coherence models are typically trained and tested on synthetic data.

LC I use the LC model detailed in §2.4.1 and §4.2.1. I conjecture that the model has an advantage and could learn complementary features to $\text{LSTM}_{\text{T\&N}}$ as it looks locally into neighbouring sentences while $\text{LSTM}_{\text{T\&N}}$ models the whole essay as one sequence, giving a more global representation of text.

MTL_{GRs} I leverage the model described in §3.2.1. As shown in §4.4.2, MTL_{GRs} is efficient in selecting maximally coherent sentence orderings from sets of candidate permutations.

MTL_{GRs}+ELMo I use the MTL_{GRs} model bootstrapped with ELMo vectors as elaborated in §3.1.1, which has further enhanced MTL performance in binary coherence assessment (§4.4.2).

MTL_{GRs}+BERT Similarly, I utilise the MTL_{GRs} model initialised with BERT vectors (§3.1.1) which has also boosted the performance of MTL in the binary coherence evaluation task.

I note that other approaches have proven their efficacy in modeling coherence such as the LCD-based ones. However, my aim is to show that supporting neural AES systems with coherence modeling could strengthen their ability to detect adversarial incoherent examples, and not to exhaustively plug all the coherence models into the joint framework to determine which model works best. Furthermore, the public implementation of the LCD models is in PyTorch while the one for LSTM_{T&N} is in Keras which requires re-implementing one of the models in the other library to facilitate their integration, which would be an interesting avenue for future work.

6.6 Joint learning

In this section, I describe my joint learning (JL) approach that combines an AES model with a coherence one. I also present different variations of the approach.

6.6.1 Approach

My main goal is to build a robust AES system that is able to correctly flag adversarial input while maintaining a high performance on essay scoring. To that end, I propose a joint learning (JL) approach that predicts a holistic essay score in addition to flagging outlier incoherent essays by integrating a coherence model with a state-of-the-art neural AES system. For the AES component, I leverage the LSTM_{T&N} and its ELMo and BERT versions, whereas for the coherence component I plug the four coherence models presented in §6.5 which results in four joint models:

- JL with LSTM_{T&N} and LC (JL-LC)
- JL with LSTM_{T&N} and MTL_{GRs} (JL-MTL_{GRs})
- JL with LSTM_{T&N}+ELMo and MTL_{GRs}+ELMo (JL-MTL_{GRs}+ELMo)

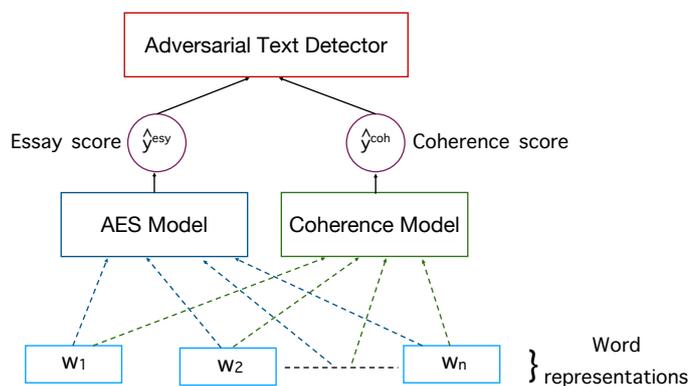


Figure 6.2: The JL framework for scoring essays as well as detecting adversarial input. The AES model is the one depicted in Fig. 6.1, and the coherence model in Fig. 4.1 or Fig. 3.2.

- JL with $\text{LSTM}_{\text{T\&N}} + \text{BERT}$ and $\text{MTL}_{\text{GRs}} + \text{BERT}$ (JL- $\text{MTL}_{\text{GRs}} + \text{BERT}$)

Training the AES and coherence models in the joint framework is straightforward. For the non-contextualised frameworks (JL-LC and JL- MTL_{GRs}), I keep the architectures of the AES and coherence models intact and train them as before but, allow them to share the word embedding layer. More specifically, starting from the same word representations, the joint model branches into an AES model that predicts an essay score (\hat{y}^{essay}) and a coherence model that predicts a coherence score (\hat{y}^{coh}). Fig. 6.2 provides an illustration for the framework. As for the JL- $\text{MTL}_{\text{GRs}} + \text{ELMo}$ and JL- $\text{MTL}_{\text{GRs}} + \text{BERT}$ models, the same framework is applied and the word embeddings are shared but kept frozen and not updated during training since they are contextualised. Accordingly, each network tunes its own set of independent parameters.

Annotation The JL framework uses the combined dataset for training which is an aggregate of both the ASAP original essays and the adversarial ones (§6.3.1). Each essay is annotated with an essay score (y^{essay}) and a coherence score (y^{coh}). During training, for the ASAP original essays, I assume that both the gold essay and coherence scores are the same and equal to the gold ASAP scores. This is not too strict an assumption, as overall scores of writing competence tend to correlate highly with overall coherence (Crossley and McNamara, 2010, 2011). For the synthetic essays, I set the “gold” coherence scores to zero, and the “gold” essay scores to those of their original non-permuted counterparts in the ASAP dataset. The intuition is as follows: firstly, if we set the “gold” essay scores of synthetic essays to zero, this might bias the model into over-predicting zeros; secondly, my approach reinforces the inability of $\text{LSTM}_{\text{T\&N}}$ to detect adversarial input, and forces the overall network to rely on the coherence branch to identify such input.

Adversarial detection The two sub-networks are trained together and the error gradients are back-propagated to the shared word embedding layer (in the non-contextualised models). In order to detect whether an essay is adversarial, I further augment the system with an *adversarial text detection* component that simply captures adversarial input based on the difference between the predicted essay and coherence scores. Specifically, I use the development set to learn a threshold¹⁰ for this difference, and flag an essay as adversarial if the difference is larger than the threshold. For simplicity, I empirically calculate the threshold as the average difference between the predicted essay and coherence scores in the synthetic data of the development set:

$$\text{threshold} = \frac{1}{M} \sum_{i=1}^M (\hat{y}_i^{esy} - \hat{y}_i^{coh}) \quad (6.5)$$

where M is the number of synthetic essays in the development set. For the final evaluation, the essays flagged as outliers by the model are assigned the minimum score in the respective prompt (most likely zero score) while the ones that pass the filter are assigned their predicted essay score as the final holistic score (\hat{y}_i^{esy}).¹¹

The description of my JL network indicates that the framework, in essence, performs MTL with two tasks: predicting an essay score and a coherence one, and the network leverages both scores to generate the final grade that is to be evaluated. Nonetheless, I use the expression ‘joint learning’ (JL) to describe this model to avoid confusion with my MTL models for coherence assessment.

6.6.2 Variations of parameter sharing

The JL models that use standard word embeddings allow sharing these embeddings between the AES and coherence sub-networks. In order to further validate this approach, I create two other variants of JL-LC, which is the benchmark JL model, with different parameter sharing setups as follows.

JL with no word embedding sharing (JL-LC_{no_layer_sharing}) In order to assess the value of sharing the word embedding layer, I create a version of the JL-LC model in which the two sub-models are trained separately without sharing the first layer of word representations. More concretely, each sub-network will have its own word embedding matrix that is only fine-tuned by this sub-network.¹²

¹⁰I note that this threshold is different than the one in Table 6.1.

¹¹This is similar to the *product of experts* approach (Hinton, 2002) used by Karimi Mahabadi et al. (2020) for debiasing by combining the predictions of two models: a base model that learns the actual task and a bias-only one that learns dataset biases and helps the base model reduce its biases.

¹²See §3.1.1 for more details about retrieving word representations from an embedding matrix.

JL with LSTM sharing (JL-LC_{lstm_sharing}) I build a version of JL-LC where the sub-models share more parameters besides word embeddings. Specifically, since both sub-networks apply an LSTM over input word vectors, I allow them to share their LSTM weights (Eq. 2.2). In other words, the same LSTM parameters are leveraged and optimised by both sub-networks. In the LSTM_{T&N} branch, LSTM is applied over the sequence of words in the entire essay, whereas in the LC branch, it is applied over the sequence of words in each sentence separately. With this approach, I hope to facilitate learning better LSTM parameters that encode relations between words across the entire essay (via LSTM_{T&N}) while also focusing on individual sentence representations (via LC). Further techniques for parameter sharing such as soft sharing (§2.5) are to be explored in future work.

6.7 Experiments

In this section, I detail my experimental setup that assesses the performance of models on essay scoring and adversarial detection. For preprocessing, all essays are lowercased and tokenised using the NLTK tokeniser.¹³ All non-contextualised models are initialised with pre-trained word embeddings (Zou et al., 2013) following Taghipour and Ng (2016). As previously mentioned in §6.3.2, during training with original or combined sets, essay scores are normalised to the range $[0, 1]$ and the predicted scores are then scaled back to the original score range to measure QWK.

$$\text{for training score} = (\text{gold score} - \text{low}) / (\text{high} - \text{low})$$

$$\text{for evaluation score} = \text{predicted score} \times (\text{high} - \text{low}) + \text{low}$$

where *low* and *high* are the minimum and maximum possible scores for the prompt that is being evaluated. For adversarial evaluation, since I perform ranking between intact and noisy essays, applying the score transformation is not important so I keep the predicted scores in the range $[0, 1]$. Training and testing are carried out for each prompt separately using 5-fold cross validation and the final evaluation is done over the test portions of the 5 folds together.

Coherence models As mentioned in §6.5, I leverage the LC, MTL_{GRs}, MTL_{GRs}+ELMo and MTL_{GRs}+BERT coherence models; I also include EGrid CNN_{ext} as a baseline. For all the models, I follow the same hyperparameter and training setup used earlier for the WSJ (§4.3). The models are trained on synthetic ASAP essays and evaluated on the synthetic test set. For evaluation, I select the models with the highest PRA value on the dev set. I

¹³<https://www.nltk.org/>

report PRA and TPRA values in Table 6.2.

AES models For LSTM_{T&N}, I replicate and evaluate the model of Taghipour and Ng (2016) using their publicly available code.¹⁴ The model is trained on original ASAP essays and tested on both the original test set using QWK (Table 6.3) and adversarial test set using PRA and TPRA metrics (Table 6.2). I create two other versions of the model where input words are initialised with ELMo or BERT (LSTM_{T&N}+ELMo and LSTM_{T&N}+BERT). Finally, I train a version of LSTM_{T&N} on combined training essays (LSTM_{T&N-comb}), where the adversarial texts are labeled with the minimum score according to the addressed prompt. For all these AES systems, I select the model that achieves the highest QWK on the dev set.

Regarding the NLI-DM-BCA model, Nadeem et al. (2019) have kindly tested their pre-trained model on my adversarial test set for prompts 1 and 2, since they only focus on these two prompts in their paper. I, therefore, limit the evaluation of NLI-DM-BCA to these prompts which should be indicative of performance on other prompts. For QWK on original essays, I report the values on the two prompts from Nadeem et al. (2019). I note that their model was trained on original essays only with no adversarial examples.

JL models For all the JL-based models, I implement the frameworks described in §6.6, where each framework combines a coherence model with an AES one, keeping the same hyperparameters as in the individual models. The thresholds for adversarial detection are estimated according to Eq. 6.5 and reported in Appendix D. The models are trained on the combined training set, and tested on both the adversarial test set (Table 6.2) and the original test set (Table 6.3).

6.8 Results

In this section, I present my results and analysis. Table 6.2 reveals the performance on adversarial detection using PRA and TPRA metrics, while Table 6.3 shows essay scoring results using QWK. I also summarise the average results in Table 6.4.

Coherence Models From Table 6.2, we can see, as expected, that the coherence models, that are trained on synthetic data only, surpass the other AES and joint models trained on original or combined datasets. Coherence models achieve average PRA > 90% with the highest performance obtained by MTL_{GRs}+ELMo and MTL_{GRs}+BERT ($\approx 98\%$). The only exception is EGrid CNN_{ext} with PRA = 72.6% suggesting the inadequacy of modeling entity transitions in learner domain in comparison to other neural coherence models that

¹⁴<https://github.com/nusnlp/nea>

Model	Training Data	PRA								
		1	2	3	4	5	6	7	8	Avg
EGrid CNN _{ext}	Synthetic	0.780	0.536	0.879	0.888	0.781	0.894	0.549	0.502	0.726
LC	Synthetic	0.960	0.978	0.944	0.965	0.943	0.964	0.887	0.934	0.946
MTL _{GRs}	Synthetic	0.988	0.985	0.949	0.927	0.957	0.976	0.949	0.975	0.963
MTL _{GRs} +ELMo	Synthetic	0.998	0.991	0.984	0.976	0.987	0.996	0.969	0.966	0.983
MTL _{GRs} +BERT	Synthetic	0.995	0.980	0.981	0.977	0.986	0.995	0.979	0.962	0.982
LSTM _{T&N}	Original	0.232	0.215	0.338	0.561	0.483	0.599	0.502	0.511	0.430
LSTM _{T&N} -comb	Combined	0.995	0.990	0.967	0.945	0.967	0.984	0.910	0.690	0.931
LSTM _{T&N} +ELMo	Original	0.651	0.692	0.277	0.651	0.403	0.345	0.520	0.758	0.537
LSTM _{T&N} +BERT	Original	0.472	0.441	0.307	0.631	0.456	0.483	0.424	0.386	0.450
NLI-DM-BCA	Original	0.218	0.147	-	-	-	-	-	-	-
JL-LC	Combined	0.932	0.791	0.728	0.683	0.736 ⁻	0.834 ⁺	0.663 ⁺	0.838 ⁺	0.775
JL-LC _{no_layer_sharing}	Combined	0.917	0.750	0.742	0.718	0.815	0.785	0.553	0.712	0.749
JL-LC _{lstm_sharing}	Combined	0.944 ⁺	0.845 ⁺	0.864 ⁺	0.807 ⁺	0.905 ⁺	0.908 ⁺	0.591 ⁺	0.309 ⁻	0.771
JL-MTL _{GRs}	Combined	0.961	0.802	0.828	0.760	0.846	0.877	0.644	0.673	0.798
JL-MTL _{GRs} +ELMo	Combined	0.927	0.542	0.852	0.759	0.875	0.892	0.807	0.595	0.781
JL-MTL _{GRs} +BERT	Combined	0.889	0.775	0.837	0.766	0.857	0.855	0.706	0.611	0.787

Model	Training Data	TPRA								
		1	2	3	4	5	6	7	8	Avg
EGrid CNN _{ext}	Synthetic	0.626	0.530	0.821	0.821	0.713	0.821	0.514	0.495	0.667
LC	Synthetic	0.571	0.636	0.729	0.867	0.704	0.881	0.544	0.583	0.689
MTL _{GRs}	Synthetic	0.986	0.981	0.951	0.925	0.962	0.976	0.950	0.972	0.962
MTL _{GRs} +ELMo	Synthetic	0.997	0.949	0.983	0.971	0.975	0.990	0.943	0.955	0.970
MTL _{GRs} +BERT	Synthetic	0.990	0.937	0.985	0.977	0.987	0.993	0.969	0.931	0.971
LSTM _{T&N}	Original	0.394	0.387	0.459	0.514	0.497	0.538	0.498	0.502	0.473
LSTM _{T&N} -comb	Combined	0.982	0.967	0.946	0.929	0.964	0.974	0.845	0.529	0.892
LSTM _{T&N} +ELMo	Original	0.507	0.517	0.475	0.531	0.491	0.475	0.500	0.510	0.500
LSTM _{T&N} +BERT	Original	0.500	0.500	0.474	0.532	0.498	0.499	0.497	0.498	0.499
NLI-DM-BCA	Original	0.377	0.281	-	-	-	-	-	-	-
JL-LC	Combined	0.933	0.791	0.730	0.690	0.737 ⁻	0.834 ⁺	0.666 ⁺	0.838 ⁺	0.777
JL-LC _{no_layer_sharing}	Combined	0.918	0.750	0.745	0.724	0.815	0.786	0.571	0.717	0.753
JL-LC _{lstm_sharing}	Combined	0.944 ⁺	0.845 ⁺	0.865 ⁺	0.812 ⁺	0.906 ⁺	0.908 ⁺	0.591	0.326 ⁻	0.774
JL-MTL _{GRs}	Combined	0.961	0.802	0.831	0.764	0.848	0.877	0.654	0.677	0.801
JL-MTL _{GRs} +ELMo	Combined	0.927	0.559	0.852	0.757	0.876	0.892	0.810	0.603	0.784
JL-MTL _{GRs} +BERT	Combined	0.891	0.767	0.837	0.768	0.858	0.856	0.711	0.621	0.788

Table 6.2: Results on the ASAP synthetic test set; the top half reports PRA and the bottom reports TPRA. Each half is horizontally divided into three partitions to visually discriminate between coherence models, AES models and JL models, in that order. For each model I display the training dataset, PRA or TPRA across the 8 prompts and in the final column I report the average across all the prompts. The superscripts + and - are added to indicate significantly better or worse (respectively) results of LSTM_{T&N}+ELMo, LSTM_{T&N}+BERT, JL-LC and JL-LC_{lstm_sharing} compared to their base models (LSTM_{T&N} for the first two and JL-LC_{no_layer_sharing} for the last two). Significance is calculated with a randomisation test for p-value < 0.01.

learn from all essay words. This is further supported by the notable drop of the PRA performance of EGrid CNN_{ext} on the learner domain vs. its obtained PRA on the news domain (87.6%), whereas all the other coherence models herein have improved over the news domain, most notably the LC model (see WSJ results in Table 4.5). Looking at the TPRA metric of the coherence models in Table 6.2, we find that the MTL approach outperforms all other models, particularly when enhanced with ELMo or BERT (average TPRA \approx 97%), while LC and EGrid CNN_{ext} do not generalise well and fall short when

Model	Training Data	QWK								Avg.
		1	2	3	4	5	6	7	8	
LSTM _{T&N}	Original	0.746	0.667	0.681	0.799	0.804	0.822	0.803	0.591	0.748
LSTM _{T&N-comb}	Combined	0.575	0.533	0.428	0.597	0.608	0.478	0.514	-0.183	0.462
LSTM _{T&N} +ELMo	Original	0.805 ⁺	0.642	0.655	0.757 ⁻	0.761 ⁻	0.782 ⁻	0.769 ⁻	0.690 ⁺	0.737
LSTM _{T&N} +BERT	Original	0.813⁺	0.650	0.660	0.763 ⁻	0.772 ⁻	0.786 ⁻	0.789	0.727⁺	0.750
NLI-DM-BCA	Original	0.800	0.671	-	-	-	-	-	-	-
JL-LC	Combined	0.769	0.648 ⁺	0.681	0.790	0.806	0.806	0.791	<u>0.506⁻</u>	0.737
JL-LC _{no.layer.sharing}	Combined	0.759	0.577	0.683	0.796	0.814	0.801	0.783	0.551	0.733
JL-LC _{lstm.sharing}	Combined	0.782 ⁺	0.601	0.661 ⁻	0.789	0.799	0.798	0.743 ⁻	0.529	0.724
JL-MTL _{GRs}	Combined	0.775	<u>0.609</u>	0.670	<u>0.763</u>	0.794	0.810	<u>0.759</u>	0.627	0.733
JL-MTL _{GRs} +ELMo	Combined	<u>0.785</u>	0.635	0.635	0.754	<u>0.781</u>	0.772	<u>0.756</u>	0.676	0.729
JL-MTL _{GRs} +BERT	Combined	<u>0.794</u>	0.642	<u>0.624</u>	0.752	<u>0.767</u>	0.780	<u>0.777</u>	0.720	0.737

Table 6.3: Results on the ASAP original test set of AES models (in the top half) and JL models (in the bottom half). For each model I display the training dataset, QWK across the 8 prompts and in the final column I report the average across all the prompts after applying Fisher transformation (§2.7). The results of NLI-DM-BCA are those reported by Nadeem et al. (2019). The superscripts + and - are added to indicate significantly better or worse (respectively) results of LSTM_{T&N}+ELMo, LSTM_{T&N}+BERT, JL-LC and JL-LC_{lstm.sharing} compared to their base models (LSTM_{T&N} for the first two and JL-LC_{no.layer.sharing} for the last two). JL-LC and the other JL MTL-based models are double-underlined or single-underlined if they are significantly better or worse (respectively) than their LSTM_{T&N} counterpart (JL-LC and JL-MTL_{GRs} vs. LSTM_{T&N}, JL-MTL_{GRs}+ELMo vs. LSTM_{T&N}+ELMo and JL-MTL_{GRs}+BERT vs. LSTM_{T&N}+BERT). Significance is calculated with a randomisation test for p-value < 0.01.

comparing coherent documents against all permuted ones in the test set, not just their incoherent counterparts. The superiority of my MTL-based models over the LC and EGrid CNN_{ext} ones is in line with my findings in Chapter 4.

AES Models Regarding the state-of-the-art AES models, we find that despite their strong performance in essay scoring, they fail to capture adversarial essays. For instance, LSTM_{T&N}, LSTM_{T&N}+ELMo and LSTM_{T&N}+BERT achieve an average QWK of 0.748, 0.737 and 0.750 respectively on the task of grading original essays (Table 6.3); however, their performance drastically drops close to chance on adversarial essay detection as shown in Table 6.2. Similarly, NLI-DM-BCA achieves outstanding performance on prompts 1 and 2 relative to the other models in Table 6.3, while yielding the poorest results in flagging outlier essays on these prompts, despite being a discourse-aware model. Even though NLI-DM-BCA is only evaluated on prompts 1 and 2, its performance on these prompts gives us an idea of its overall adequacy for essay scoring and adversarial identification. Furthermore, PRA and TPRA for NLI-DM-BCA are much lower than chance; this is because a large number of its predictions are ties; i.e., an original essay and its shuffled version are assigned the same grade, which is marked as an incorrect prediction with the pairwise evaluation measures. Nevertheless, I note that LSTM_{T&N}, LSTM_{T&N}+ELMo, LSTM_{T&N}+BERT and NLI-DM-BCA are strictly trained on original essays and they only see adversarial

examples during testing. When $\text{LSTM}_{\text{T\&N}}$ is trained on the combined dataset of original and adversarial essays (i.e., $\text{LSTM}_{\text{T\&N-comb}}$), its performance significantly increases on adversarial detection (PRA= 0.931 and TPRA= 0.892), yet significantly drops on essay scoring (QWK= 0.462). It would be interesting to evaluate the NLI-DM-BCA model when trained on the combined dataset, which I leave to future work. In summary, the results show that current state-of-the-art AES systems are not well-suited at capturing adversarially-crafted input of grammatical albeit incoherent sequences of sentences, which needs to be addressed to ensure the validity of these systems.

I now analyse the performance when utilising contextualised embeddings for essay scoring in Table 6.3. When $\text{LSTM}_{\text{T\&N}}$ is initialised with BERT vectors ($\text{LSTM}_{\text{T\&N+BERT}}$), it achieves the best overall performance (average QWK = 0.750), yet performs closely to its base $\text{LSTM}_{\text{T\&N}}$ model that uses standard vectors (average QWK = 0.748). We notice that while $\text{LSTM}_{\text{T\&N+BERT}}$ significantly outperforms $\text{LSTM}_{\text{T\&N}}$ on two prompts (1 and 8), the latter significantly outperforms the former on three prompts (4, 5 and 6). As for $\text{LSTM}_{\text{T\&N+ELMo}}$, overall, it performs slightly worse than $\text{LSTM}_{\text{T\&N}}$ (average QWK = 0.737), but with a significant improvement on prompts 1 and 8, and a significant drop on prompts 4, 5, 6 and 7. Additionally, we notice that prompt 8 is the prompt that gains the most from adding ELMo or BERT (with $\approx 10\%$ improvement with ELMo and $\approx 13\%$ with BERT), which could be attributed to the small size of its data (723 essays) and the ability of contextualised embeddings to help with low-resource tasks. Regarding the prompts where there is a significant drop with contextualised vectors, it is not clear why this happens but, a good investigation starting point is to examine other approaches to creating ELMo or BERT embeddings (i.e., use different combinations of layers). For instance, Reimers and Gurevych (2019) explored different combining methods for ELMo layers for the task of detecting arguments in persuasive essays, and found that learning a weighted average of the two lowest layers achieves the best performance and outperforms calculating a weighted average of the three layers (Peters et al., 2018) or leveraging the top layer (Peters et al., 2017), which I use throughout this thesis. Since my aim is not to build a state-of-the-art AES system or to find the best contextualised embeddings for the task, I leave this investigation to future work.

JL Models Moving to JL analysis, my aim is to investigate the efficacy of the JL approach on essay scoring (Table 6.3) as well as identifying adversarial input (Table 6.2). As for the latter task, the results show that the JL models significantly outperform other AES models, with the exception of $\text{LSTM}_{\text{T\&N-comb}}$ that is trained on combined data yet completely fails at essay scoring. JL models, on the other hand, are capable of capturing adversarial essays, while maintaining competitive performance on essay scoring as reported in Table 6.3. For better visualisation of the performance on both tasks, I plot in Fig. 6.3

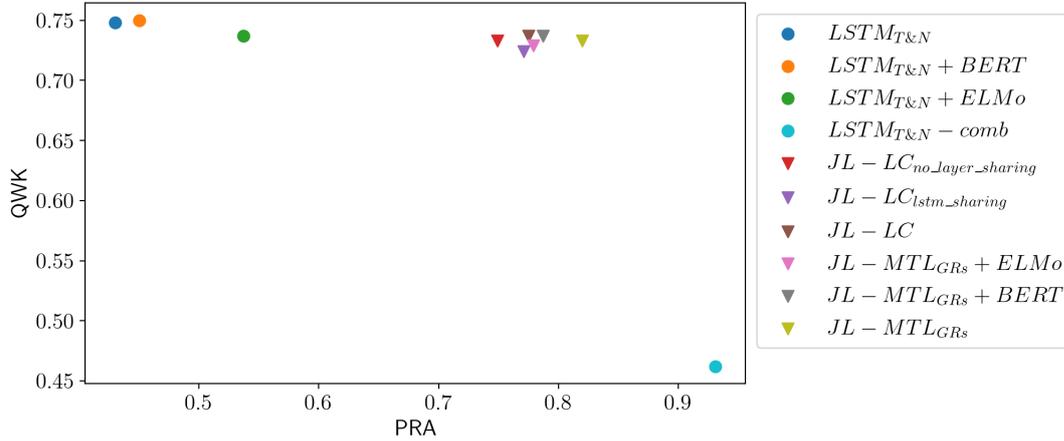


Figure 6.3: The graph represents the performance of the models on essay scoring (QWK on the y-axis) and adversarial detection (PRA on the x-axis). The JL models are represented by triangles and the AES ones by circles.

the PRA and QWK values achieved by the models. The figure reveals that the JL models (represented by triangles) achieve high performance on essay scoring and adversarial detection, whereas the AES models (represented by circles) either succeed in essay scoring but fail in adversarial detection ($LSTM_{T\&N}$, $LSTM_{T\&N}+ELMo$ and $LSTM_{T\&N}+BERT$) or vice versa ($LSTM_{T\&N}-comb$). The $LSTM_{T\&N}-comb$ and JL comparison further supports my hypothesis that forcing the JL models to rely on the coherence branch for adversarial input detection contributes to building more reliable AES systems. We need something more than just training a state-of-the-art AES model (in my case, $LSTM_{T\&N}$) on both original and synthetic data.

In order to investigate the value of parameter sharing, I compare the basic $JL-LC_{no_layer_sharing}$, that does not allow any parameter sharing between the two sub-networks, with its counterparts: $JL-LC$ that shares the word embedding layer and $JL-LC_{lstm_sharing}$ that shares both the word embedding and word LSTM layers. We find that, on adversarial detection, $JL-LC$ outperforms $JL-LC_{no_layer_sharing}$ on both PRA and TPRA measures, with a significant improvement on three prompts (6, 7 and 8) but also a significant drop on prompt 5. On the other hand, $JL-LC_{lstm_sharing}$ also, overall, surpasses $JL-LC_{no_layer_sharing}$, with a constant significant increase on the first 7 prompts (with the exception of TPRA on prompt 7 where the increase is not significant), yet a drastic drop on prompt 8 (the prompt with the lowest representation). As for the performance on essay assessment, by looking at average QWK, we find that $JL-LC$ and $JL-LC_{lstm_sharing}$ perform closely to the base $JL-LC_{no_layer_sharing}$. On closer inspection, we find that $JL-LC$ significantly outperforms $JL-LC_{no_layer_sharing}$ on prompt 2 and significantly underperforms it on prompt 8, while $JL-LC_{lstm_sharing}$ significantly outperforms $JL-LC_{no_layer_sharing}$ on prompt 1 but underperforms it on prompts 3 and 7. In summary, the comparison between $JL-LC$ models suggests that adversarial detection benefits more from parameter sharing while this sharing

Model	Synthetic Test		Original Test	
	PRA	TPRA	QWK	Error
LSTM _{T&N}	0.430	0.473	0.748	-
LSTM _{T&N-comb}	0.931	0.892	0.462	-
LSTM _{T&N} +ELMo	0.537	0.500	0.737	-
LSTM _{T&N} +BERT	0.450	0.499	0.750	-
JL-LC	0.775	0.777	0.737	0.38%
JL-LC _{no_layer_sharing}	0.749	0.753	0.733	0.64%
JL-LC _{lstm_sharing}	0.771	0.774	0.724	0.43%
JL-MTL _{GRs}	0.820	0.822	0.733	0.36%
JL-MTL _{GRs} +ELMo	0.779	0.782	0.729	0.56%
JL-MTL _{GRs} +BERT	0.787	0.788	0.737	0.54%

Table 6.4: Summary of average PRA and TPRA on the synthetic test set and average QWK on the original test set. The last column reports the average percentage error of flagging an original essay as adversarial in the original test set.

has a modest effect on essay scoring.

I now compare my JL approach to the LSTM_{T&N} one on essay scoring using QWK (Table 6.3). More concretely, I compare each JL approach with its LSTM_{T&N} counterpart, that is: JL-LC and JL-MTL_{GRs} vs. LSTM_{T&N}, JL-MTL_{GRs}+ELMo vs. LSTM_{T&N}+ELMo, and JL-MTL_{GRs}+BERT vs. LSTM_{T&N}+BERT. I find that there is a minor drop in the overall performance (average QWK) of every JL model vs. its AES counterpart. By examining QWK on each prompt, we find that JL-LC performs closely to LSTM_{T&N} on the first 7 prompts, but significantly underperforms it on prompt 8. On the other hand, JL-MTL_{GRs} and JL-MTL_{GRs}+BERT perform closely to their AES counterparts on 5 prompts but perform significantly worse on the other 3. Finally, JL-MTL_{GRs}+ELMo is close to LSTM_{T&N}+ELMo on 5 prompts, significantly outperforms it on prompt 5 and significantly underperforms it on prompts 1 and 7.

In general, JL models maintain competitive performance relative to the LSTM-based AES systems on most of the prompts in the task of predicting a holistic score for essays, while significantly boosting the performance on adversarial detection on all prompts. Among the JL family of models, JL-LC provides the most stable performance on essay scoring, in comparison to its AES counterpart and only drops significantly on prompt 8, which has the lowest data representation in the ASAP dataset. Nevertheless, overall, all JL models (JL-LC and the ones that use MTL) perform competitively with their AES counterparts on essay scoring, drop on maximum 3 prompts out of 8 and could potentially achieve a significant increase (JL-MTL_{GRs}+ELMo on prompt 5). Therefore, the JL framework is a promising approach to more reliable and robust essay scoring systems.

Further Analysis Ideally, no essays in the ASAP original data should be flagged as adversarial as they were not written to trick the system. I, therefore, calculate the number

of original texts incorrectly detected as adversarial, and report the average error across prompts for all JL models in Table 6.4. We find that the error percentage is quite small ($< 0.65\%$ for all the models), particularly with JL-LC and JL-MTL_{GRs} (the latter has the lowest error of 0.36%). This small error further promotes JL as a reliable approach to essay scoring.

Finally, I investigate both the essay and coherence scores predicted by the JL models for the permuted and original essays in the synthetic dataset and depict the scores for prompts 1 and 8 in Fig. 6.4. I select prompt 1 as it has the highest overall results in adversarial detection and 8 as it has the lowest, for JL models (Table 6.2). The Figure shows a large difference between predicted essay and coherence scores on adversarial essays, for all the JL models (1./8. b, d, f and h), where the models predict high essay scores for permuted texts (as a result of my training and annotation strategy), but low coherence scores (as predicted by the coherence sub-network). For highly scored ASAP original essays (1./8. a, c, e and g), the predictions are less varied and positively contribute to the performance of my proposed JL approach. This distinction between essay and coherence scores for original and adversarial essays show the efficacy of my JL approach and the ability of the adversarial detection component to discriminate between coherent and adversarial essays based on their predicted coherence and essay scores (Fig. 6.2). Furthermore, the results motivate my annotation strategy, where for original essays, I assume that coherence scores are equal to essay scores, whereas for synthetic essays, I assume coherence scores to be equal to the lowest possible score and essay scores to be those of their original counterparts.

6.9 Summary

In this chapter, I have demonstrated that state-of-the-art approaches to AES are not well-suited to capturing adversarially crafted input of grammatical but incoherent sequences of sentences. I, therefore, have developed a JL framework that combines a neural AES model with a coherence one to simultaneously predict a holistic score for essays and flag adversarial incoherent input. My JL approach significantly enhances the ability of the AES system to detect adversarial input while maintaining a competitive performance in predicting a holistic essay score, providing a promising research direction that ensures the validity of AES systems. This is particularly true for the JL-LC framework that integrates the state-of-the-art LSTM_{T&N} AES model with a local coherence (LC) model as the network branch that checks for coherence. I found that JL-LC performs closely to LSTM_{T&N} on essay scoring on all the ASAP prompts, with the exception of prompt 8 that has the smallest number of essays. I have also studied the effect of adding contextualised embeddings to LSTM_{T&N} and found that they are helpful with low-resource prompts

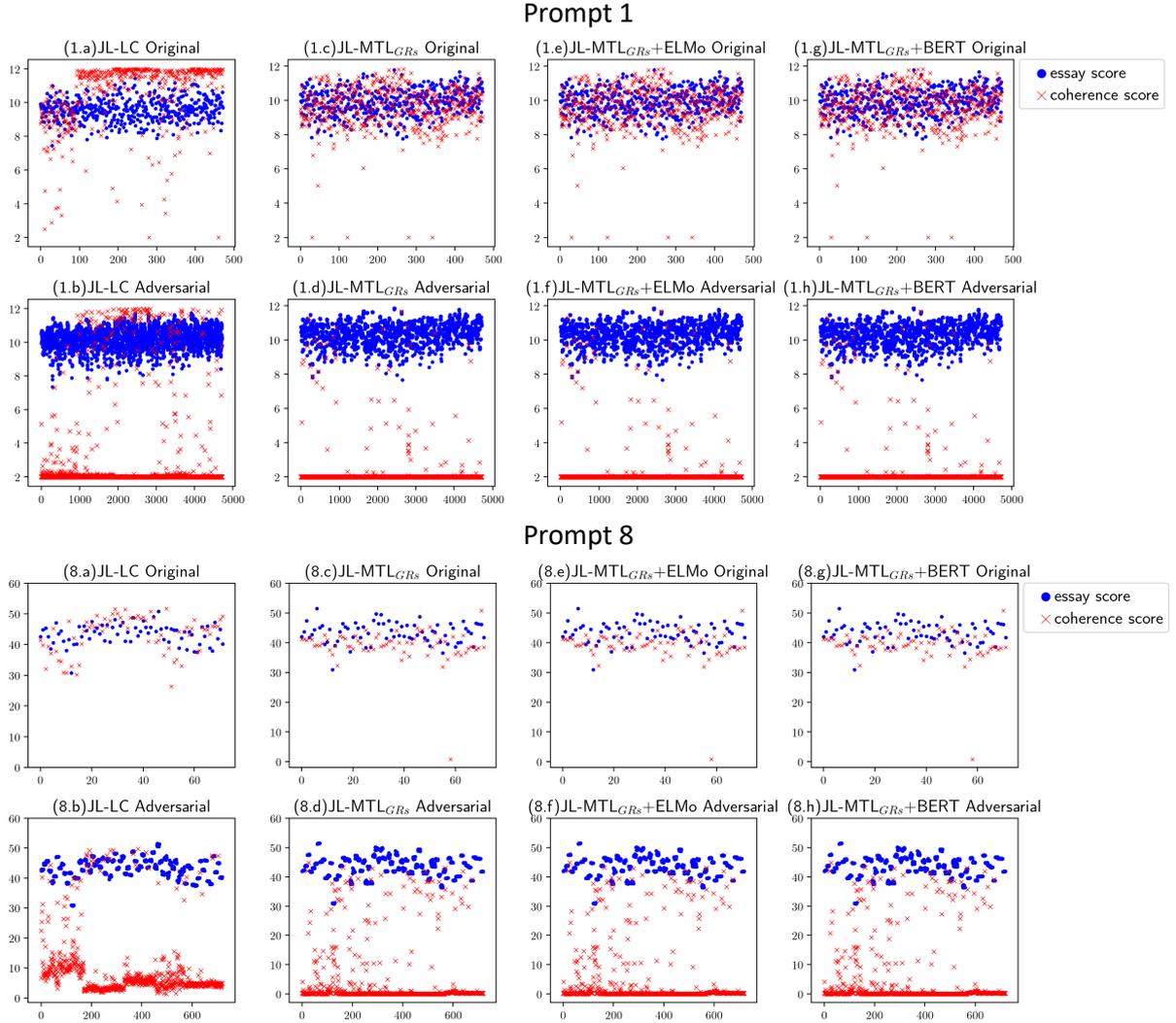


Figure 6.4: Predictions of the JL models on the synthetic test set for prompts 1 (first two rows) and prompt 8 (last two rows). I show results for JL-LC, JL-MTL_{GRs}, JL-MTL_{GRs}+ELMo and JL-MTL_{GRs}+BERT. The upper graphs for each prompt ((a), (c), (e) and (g)) show the predicted essay and coherence scores on original essays, whereas the bottom ones ((b), (d), (f) and (h)) show the predicted scores for highly scored original ASAP essays (y-axis represents scores). The blue circles represent essay scores, and the red crosses represent coherence scores. All predicted scores are mapped to their original scoring scale.

(i.e., prompt 8) but, overall, they do not provide further gains over standard embeddings. This motivates examining other approaches to building contextualised embeddings such as interpolating different layers of the embeddings. Furthermore, I investigated the effect of sharing different layer parameters in my JL-LC framework and found that adversarial detection benefits from parameter sharing while this sharing has a small effect in essay scoring. The overall best parameter sharing setup is achieved when the word embedding layer is shared between the two JL sub-networks.

CONCLUSION

7.1 Summary and findings

This thesis contributes to the research on discourse coherence from three perspectives: modeling, evaluation and application.

Modeling In Chapter 3, I presented my neural discriminative approach to coherence modeling. I proposed a hierarchical neural network where a Bi-LSTM with attention was utilised to build textual representations at different levels of the network. The network is trained in an MTL fashion where the network predicts a document-level coherence score at its output layer (the main task) together with word-level syntactic properties at lower layers (the auxiliary task), exploiting the hierarchical inductive transfer between the two tasks. MTL is an efficient learning approach as the use of syntactic parsers is limited to training time. I examined two types of syntactic properties: GRs and POS tags. Furthermore, I investigated the impact of initialising the model with contextualised embeddings (ELMo or BERT). Moreover, to validate my MTL approach, I compared it to STL where the same hierarchical network only performs one task of predicting a coherence score without leveraging syntactic information. I also compared it to the same network that incorporates syntactic features in different fashions such as concatenating them to input word vectors, only predicting subject and object roles as the secondary task or combining two auxiliary objectives: one for GRs and the other for POS tags.

I evaluated and analysed the performance of the models in Chapter 4. The models were tested on two domains of discourse coherence: a) a binary synthetic domain where coherent WSJ articles were compared to their noisy counterparts, created by randomly shuffling their sentences and b) the realistic domain of everyday writing (forum posts and emails), where the documents were annotated with low, medium or high levels of coherence. In the synthetic domain, I used two evaluation metrics: PRA that ranks an original coherent text

against its permuted versions and TPRA which is a stricter measurement that compares each coherent text against all the incoherent documents in the dataset. In addition, I compared MTL to state-of-the-art neural coherence models. Specifically, I included models that leverage a CNN operating on EGrid representations of text, a model that captures local coherence (LC) by scoring local text cliques and a local coherence discriminator (LCD) that encodes sentences then adds a discriminative layer over each sentence pair to distinguish between coherent and incoherent pairs. For LCD, I experimented with four different encoders: a generative RNN language model that is the current published state-of-the-art on the WSJ, and averaged fastText, ELMo or BERT embeddings.

My experiments showed the efficacy of MTL in the synthetic domain, with a significant boost when enhanced with contextualised embeddings. MTL (with GRs or POS tags) + ELMo or BERT achieved state-of-the-art results on the TPRA metric (96.9%), while LCD-BERT yielded state-of-the-art PRA (97.1%) but performed significantly worse on TPRA (92.2%). Furthermore, there was a substantial improvement of MTL over STL or models that leverage syntactic features either by concatenating them to input word vectors, solely focusing on subject and object prediction in MTL, or using random/wrong syntactic labels as the gold labels for the auxiliary objective in MTL. The superiority of MTL over such models further demonstrates the strength of the approach. However, my empirical evaluation revealed that there is negligible difference between utilising GRs or POS tags as secondary labels and no performance gains from combining both types of syntactic labels; the discrepancy between initialising with ELMo or BERT embeddings is also negligible. With further analysis and by visualising attention weights, I found that MTL models tend to focus on subject words, particularly when they utilise a GR-based objective, corroborating Centering theory and reflecting the nature of the dataset.

On close inspection to the results on the synthetic domain, I found that high performing models exhibit some ability to capture partial coherence, which could be better examined in a realistic domain of writing. I, therefore, extended my evaluation to a more realistic dataset that exhibits various degrees of coherence and compared MTL to the published state-of-the-art models on this dataset: LC and a hierarchical LSTM (with no attention). When I tested the models on identifying discrete classes of coherence, I found that MTL with a GR auxiliary (MTL_{GRs}) achieved the best overall performance measured as the average accuracy (58.0%) over the three realistic datasets: Yahoo, Clinton and Enron. Nonetheless, even the strongest models failed at recognising texts of medium coherence level. This could be attributed to the small size of the dataset and the difficulty and subjectivity of assessing various degrees of coherence in comparison to binary data, particularly with medium level documents that are more difficult for humans to agree on. These results were also supported by attention visualisation that suggested that coherence features are less pronounced in realistic data.

Additionally, applying transfer learning from the WSJ to the realistic domain negatively affected the performance on Yahoo and Enron and only achieved a 1% gain on Clinton, which could be explained by the different natures of the source and target domains. Finally, for a less strict evaluation, I conducted experiments where I cast the task as a ranking one and used Spearman’s correlation for evaluation. With this new setup, BERT-based models yielded the best performance followed by ELMo, while MTL_{GRs} fell behind further showing the ability of contextualised embeddings to capture more fine-grained ranks. Unlike synthetic data, the superiority of BERT over ELMo is more noticeable in the realistic domain. Assessing synthetically created documents is more straightforward and potentially less effective in highlighting the difference between different contextualised embeddings as they all have the power to solve the task.

Evaluation In Chapter 5, I proposed an evaluation framework to investigate the linguistic aspects implicated in discourse organisation that coherence models capture. To that end, I devised two datasets: (1) CCD which is a large-scale out-of-training-domain dataset that examines the robustness of models against semantic and syntactic changes that result in less coherent texts and (2) CLAD which is a small-scale test dataset of more controlled semantic and syntactic perturbations from a domain similar to the WSJ. The results on both datasets revealed that RNN-based models tended to memorise syntactic patterns that co-occur in coherent texts but were unable to capture other semantic or rhetorical features. The models benefited from further fine-tuning on the CCD, yet there was still a substantial gap between their ability to detect syntactic corruptions vs. semantic ones. On the other hand, the LCD approach that utilises a sentence encoder that averages word representations was more able to detect semantic properties. Additionally, results on the CLAD showed that the models did not rely on positional features if the main syntactic construction was maintained. The results also illustrated that RNN models were not sensitive to minor lexical perturbations nor could they resolve pronouns, whereas LCD-BERT achieved the most promising results on these problems. My evaluation, specifically on syntactic tasks, revealed that on both the CCD and CLAD, pre-training the models on the WSJ consistently outperformed pre-training on the realistic domain (Yahoo). This could be attributed to the nature of these domains; the WSJ exhibits syntactic regularities in its original documents that are broken in the shuffled ones, enabling the model to capture syntax.

Application In Chapter 6, I presented an application for coherence models in the pedagogical domain. I first demonstrated that state-of-the-art neural approaches to AES (even discourse-aware ones) are vulnerable to adversarially crafted input of grammatical but incoherent sequences of sentences. Accordingly, I proposed a framework for integrating and jointly training a discourse model with a state-of-the-art neural AES system in order

to enhance its ability to detect such adversarial input; the framework is trained on a combination of original essays and adversarial ones. I experimented with integrating an LSTM-based AES system with four coherence models: LC, MTL_{GRs} , $MTL_{GRs}+ELMo$ and $MTL_{GRs}+BERT$. My joint learning approach significantly enhanced the ability of the AES system to flag adversarial input while maintaining a competent performance in predicting a holistic essay score, contributing to the development of an approach that strengthens the validity of neural AES models. In particular, utilising the LC model as the coherence branch in the framework significantly improved the ability of the AES system to detect adversarial input on all the prompts, as well we maintaining a high performance at holistic score prediction across all the prompts, with the exception of prompt 8 that has the smallest number of essays. I also experimented with different parameter sharing setups for joint learning with LC, and observed that parameter sharing was useful for adversarial detection, but had a small effect in essay scoring, and that the overall best setup shared the word embedding layer between the two sub-networks (AES and LC). Finally, the experiments in Chapter 6 revealed that using contextualised embeddings did not yield further gains over standard embeddings except with low-resource prompts.

7.2 Future work

The work presented in this thesis inspires various directions for future research as follows.

Datasets My findings in Chapter 5 demonstrated the ability of neural coherence models to capture syntactic patterns while falling short in understanding the underlying semantics. This motivates the revision of the traditional methods used to create artificial coherence datasets. For example, instead of just permuting the sentence order in coherent texts, the noisy examples could be generated by maintaining the main syntactic patterns while changing semantics, in a way similar to the random examples in the CLAD (§5.3) or cloze_rand dataset (§5.2.2). Generating synthetic examples from multiple documents would be a good starting point. A similar approach was adopted by Wang et al. (2017b) who created a dataset of text pairs from different documents to train a neural network to rank semantic coherence between text segments for topic segmentation. The network was trained to rank text pairs from the same paragraph higher than pairs that are from different paragraphs but belong to the same document which in turn should be scored higher than pairs extracted from different documents. I can create artificial documents from multiple sources and constrain the creation process to preserve the main syntactic structure of the original document. This way, models could be forced to focus on semantic properties and their capacity to capture meaning will be tested.

Furthermore, my analysis in §4.4.3.1 suggests some ability for the models to capture

partial orderings which I want to further investigate in the future. This will be achieved with one of two training approaches:

- Instead of labeling the incoherent documents with zero, we annotate them with their Kendall’s τ score that indicates their similarity to the original documents and train the models to predict these multiple ranks (Feng and Hirst, 2012).
- We apply local partial permutations to the documents to create different levels of incoherence (Moon et al., 2019).

These approaches will however require human annotations (even on a sample of documents) to ensure their validity.

Investigating new methods to create and annotate artificial data is not the only direction that would benefit coherence modeling research. The fair human agreement on the GCDC (Table 4.2) motivates further investigation with regards to how to address data annotation in the realistic domain. The fair agreement was probably due to the general guidelines given to the annotators in addition to the subjectivity of the task. Future work might thus investigate how to improve the data annotation process in order to raise inter-rater agreement.

MTL auxiliary functions In this thesis, I presented two types of auxiliary functions for coherence assessment: GR prediction spurred by Centering theory and POS tags prediction to model intentional discourse structure. In the future, it would be interesting to examine other coherence-relevant auxiliary tasks in my MTL approach. Particularly, I want to predict coherence/rhetorical relations between textual units. As I mentioned in §2.5, Jernite et al. (2017) used a similar auxiliary objective that detects the type of explicit coherence relation between two sentences. This was achieved by selecting pairs of sentences, where the second sentence starts with one of pre-defined phrases belonging to specific discourse categories, removing these phrases, and training the model to predict the category of the removed phrases. They did not however, test their approach on coherence assessment. I could thus leverage their auxiliary function in my MTL framework and extend it to exploit implicit coherence relations. The WSJ corpus is a good candidate to use as it has released rhetorical relation annotations (RST-DT) (Carlson et al., 2001).

Transfer learning and parameter sharing In this thesis, I leveraged transfer learning and particularly hard parameter sharing between different tasks and domains. In §4.5.2.2, I conducted an experiment where I pre-trained a neural network on the WSJ then fine-tuned it on the GCDC. In Chapter 6, I investigated word embedding and/or LSTM sharing between the sub-networks in the joint learning framework to perform essay scoring and adversarial detection. I would like to examine other transfer learning and parameter

sharing approaches such as adaptive learning rates (Howard and Ruder, 2018), or soft parameter sharing (Ruder, 2017). I would also like to conduct cross-domain experiments between the different GCDC datasets.

Other sentence encoders I have focused in this thesis on evaluating and analysing RNN-based approaches (STL and MTL and their variations, LCD-L and LC) and comparing them to other neural entity-grid approaches or models that average word vectors. It would be interesting to study other sentence encoders such as transformer networks and compare their performance and the features they capture to sequential models. It is also worth investigating the effect of fine-tuning pre-trained contextualised encoders to be more task-specific; i.e., fine-tuning ELMo and BERT models in LCD or MTL models instead of just leveraging one untuned layer from their pre-trained representations. My work has also focused on discriminative approaches and I would like to apply my framework in Chapter 5 on the generative approaches discussed in §2.4.2 and observe how they differ from discriminative models. Finally, I only focused on models that do not use pairwise training strategy in the GCDC experiments, but in the future, I would like to adapt the LCD and neural EGrid models to multi-class prediction tasks.

Adversarial training As for my work in Chapter 6 on adversarial detection, I would like first to train discourse-aware models (e.g., NLI-DM-BCA) on the combined dataset and observe the change in their performance on essay scoring and adversarial identification. Furthermore, I am interested to investigate new ways of selecting the threshold value used in the adversarial detector of the joint learning model (§6.6) such as learning this threshold automatically while training the network, as a network parameter, then applying it at test time. Finally, I would like to evaluate my joint learning approach on the adversarial examples generated by Kumar et al. (2020b) (given that they publish their dataset). I expect it will fail on the examples that include perturbations not directly related to coherence (such as grammatical errors); however, it would be interesting to test it on discourse-related examples such as adding random sentences from leader speeches or song lyrics to student responses.

Application in other domains Finally, I would like to explore other domains for discourse coherence. In §1.3, I referred to the utility of coherence models in the mental health domain, such as detecting schizophrenic discourses. I would like to extend my models to this domain as it is an important real-life application to discourse models. My research will particularly focus on adapting the models to this low-resource task as data collection is expensive in this field, which is why neural approaches have been under-investigated in this domain.

BIBLIOGRAPHY

- Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. <https://openreview.net/forum?id=BJh6Ztuxl>. Cited on page 108.
- Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*. pages 1638–1649. Cited on page 37.
- Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic Text Scoring Using Neural Networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 715–725. <https://www.aclweb.org/anthology/P16-1068>. Cited on pages 50, 120, and 121.
- Øistein E Andersen, Helen Yannakoudakis, Fiona Barker, and Tim Parish. 2013. Developing and testing a self-assessment and tutoring system. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications, BEA*. Association for Computational Linguistics, pages 32–41. <https://www.aclweb.org/anthology/W13-1704>. Cited on page 120.
- Nicholas Asher and Alex Lascarides. 2003. Logics of conversation . Cited on page 64.
- Yigal Attali and Jill Burstein. 2006. Automated Essay Scoring With e-rater® V.2. *The Journal of Technology, Learning and Assessment* 4(3). <https://ejournals.bc.edu/index.php/jtla/article/view/1650>. Cited on pages 120 and 122.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. <http://arxiv.org/abs/1409.0473>. Cited on pages 39 and 48.

- Pierre Baldi and Peter Sadowski. 2013. Understanding Dropout. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. Curran Associates Inc., USA, NIPS'13, pages 2814–2822. <http://dl.acm.org/citation.cfm?id=2999792.2999926>. Cited on page 74.
- Yujia Bao, Shiyu Chang, Mo Yu, and Regina Barzilay. 2018. Deriving Machine Attention from Human Rationales. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, pages 1903–1913. <https://www.aclweb.org/anthology/D18-1216>. Cited on page 48.
- Megan S. Barker, Breanne Young, and Gail A. Robinson. 2017. Cohesive and coherent connected speech deficits in mild stroke. *Brain and Language* 168:23 – 36. <http://www.sciencedirect.com/science/article/pii/S0093934X16300025>. Cited on page 19.
- Regina Barzilay and Michael Elhadad. 1997. Using Lexical Chains for Text Summarization. In *Intelligent Scalable Text Summarization*. <https://www.aclweb.org/anthology/W97-0703>. Cited on pages 16, 20, and 29.
- Regina Barzilay and Noemie Elhadad. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research* 17:35–55. Cited on page 20.
- Regina Barzilay and Mirella Lapata. 2005. Modeling Local Coherence: An Entity-Based Approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Association for Computational Linguistics, Ann Arbor, Michigan, pages 141–148. <https://www.aclweb.org/anthology/P05-1018>. Cited on pages 16, 17, 18, 31, and 49.
- Regina Barzilay and Mirella Lapata. 2008. Modeling Local Coherence: An Entity-Based Approach. *Computational Linguistics* 3(1):1–34. https://people.csail.mit.edu/regina/my_papers/coherence.pdf. Cited on pages 17, 18, 19, 20, 27, 31, 32, 33, 62, 68, 82, and 99.
- Regina Barzilay and Lillian Lee. 2004. Catching the Drift: Probabilistic Content Models, with Applications to Generation and Summarization. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*. Association for Computational Linguistics, Boston, Massachusetts, USA, pages 113–120. <https://www.aclweb.org/anthology/N04-1015>. Cited on pages 19, 20, 30, 31, 49, and 99.

- Regina Barzilay and Kathleen R McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics* 31(3):297–328. Cited on page 20.
- Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. 2018. Evaluating Discourse Phenomena in Neural Machine Translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, pages 1304–1313. <https://www.aclweb.org/anthology/N18-1118>. Cited on page 20.
- Gillinder Bedi, Facundo Carrillo, Guillermo A Cecchi, Diego Fernández Slezak, Mariano Sigman, Natália B Mota, Sidarta Ribeiro, Daniel C Javitt, Mauro Copelli, and Cheryl M Corcoran. 2015. Automated analysis of free speech predicts psychosis onset in high-risk youths. *npj Schizophrenia* 1:15030. Cited on pages 19 and 29.
- Isaac I. Bejar, Michael Flor, Yoko Futagi, and Chaintanya Ramineni. 2014. On the vulnerability of automated scoring to construct-irrelevant response strategies (CIRS): An illustration. *Assessing Writing* 22:48 – 59. <http://www.sciencedirect.com/science/article/pii/S1075293514000257>. Cited on page 122.
- Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and Natural Noise Both Break Neural Machine Translation. In *6th International Conference on Learning Representations, ICLR 2018*. Vancouver, BC, Canada. <https://openreview.net/forum?id=BJ8vJebC->. Cited on pages 47 and 48.
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3(Feb):1137–1155. Cited on page 36.
- Mine Berker and Tunga Güngör. 2012. Using Genetic Algorithms with Lexical Chains for Automatic Text Summarization. In *ICAART (1)*. pages 595–600. Cited on page 29.
- Michael W Berry, Susan T Dumais, and Gavin W O’Brien. 1995. Using linear algebra for intelligent information retrieval. *SIAM review* 37(4):573–595. Cited on page 29.
- William Blacoe and Mirella Lapata. 2012. A Comparison of Vector-based Representations for Semantic Composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, Jeju Island, Korea, pages 546–556. <https://www.aclweb.org/anthology/D12-1050>. Cited on page 39.

- Daniel Blanchard, Joel Tetreault, Derrick Higgins, Aoife Cahill, and Martin Chodorow. 2013. TOEFL11: A corpus of non-native English. *ETS Research Report Series* 2013(2):i–15. Cited on page 127.
- David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3(Jan):993–1022. Cited on page 43.
- Terra Blevins, Omer Levy, and Luke Zettlemoyer. 2018. Deep RNNs Encode Soft Hierarchical Syntax. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Melbourne, Australia, pages 14–19. <https://doi.org/10.18653/v1/P18-2003>. Cited on page 102.
- J Kathryn Bock and Carol A Miller. 1991. Broken agreement. *Cognitive psychology* 23(1):45–93. Cited on page 102.
- Tanner Bohn, Yining Hu, Jinhang Zhang, and Charles Ling. 2019. Learning Sentence Embeddings for Coherence Modelling and Beyond. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*. pages 151–160. Cited on pages 17 and 45.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5:135–146. Cited on page 37.
- Danushka Bollegala, Naoaki Okazaki, and Mitsuru Ishizuka. 2006. A Bottom-Up Approach to Sentence Ordering for Multi-Document Summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Sydney, Australia, pages 385–392. <https://doi.org/10.3115/1220175.1220224>. Cited on page 19.
- Leo Born, Mohsen Mesgar, and Michael Strube. 2017. Using a Graph-based Coherence Model in Document-Level Machine Translation. In *Proceedings of the Third Workshop on Discourse in Machine Translation*. pages 26–35. Cited on page 20.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 632–642. <https://www.aclweb.org/anthology/D15-1075>. Cited on page 42.

- Susan E. Brennan, Marilyn W. Friedman, and Carl J. Pollard. 1987. A Centering Approach to Pronouns. In *Proceedings of the 25th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '87, pages 155–162. <https://doi.org/10.3115/981175.981197>. Cited on pages 27 and 28.
- Gillian Brown and George Yule. 1983. *Discourse analysis*. Cambridge university press. Cited on page 24.
- Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics* 19(2):263–311. <https://www.aclweb.org/anthology/J93-2003>. Cited on page 30.
- Meru Brunn, Yllias Chali, and Christopher J Pinchak. 2001. Text summarization using lexical chains. In *Proc. of Document Understanding Conference*. Citeseer. Cited on page 29.
- Jill Burstein, Martin Chodorow, and Claudia Leacock. 2003. Criterion: Online essay evaluation: An application for automated evaluation of student essays. In *Proceedings of the fifteenth annual conference on innovative applications of artificial intelligence*. American Association for Artificial Intelligence, pages 3–10. <https://www.aaai.org/Papers/IAAI/2003/IAAI03-001.pdf>. Cited on page 120.
- Jill Burstein, Karen Kukich, Susanne Wolff, Chi Lu, Martin Chodorow, Lisa Braden-Harder, and Mary Dee Harris. 1998. Automated Scoring Using a Hybrid Feature Identification Technique. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*. Association for Computational Linguistics, USA, page 206–210. <https://doi.org/10.3115/980845.980879>. Cited on pages 31 and 122.
- Jill Burstein, Joel Tetreault, and Slava Andreyev. 2010. Using Entity-Based Features to Model Coherence in Student Essays. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 681–684. <https://www.aclweb.org/anthology/N10-1099>. Cited on pages 17, 18, 19, 99, and 120.
- Jill Burstein, Joel Tetreault, and Martin Chodorow. 2013. Holistic discourse coherence annotation for noisy essay writing. *Dialogue & Discourse* 4(2):34–52. Cited on pages 18, 33, and 91.

- Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*. <https://www.aclweb.org/anthology/W01-1605>. Cited on pages 26 and 145.
- Patricia L Carrell. 1982. Cohesion is not coherence. *TESOL quarterly* 16(4):479–488. Cited on page 24.
- Rich Caruana. 1997. Multitask learning. *Machine learning* 28(1):41–75. Cited on pages 18 and 45.
- Richard Caruana. 1993. Multitask Learning: A Knowledge-Based Source of Inductive Bias. In *Proceedings of the Tenth International Conference on Machine Learning*. Morgan Kaufmann, pages 41–48. Cited on page 46.
- Wallace Chafe. 1976. Givenness, contrastiveness, definiteness, subjects, topics, and point of view. *Subject and topic* . Cited on page 27.
- Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*. ISCA, pages 2635–2639. http://www.isca-speech.org/archive/interspeech_2014/i14_2635.html. Cited on page 38.
- Danqi Chen and Christopher Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pages 740–750. <https://www.aclweb.org/anthology/D14-1082>. Cited on pages 61, 62, and 75.
- Erdong Chen, Benjamin Snyder, and Regina Barzilay. 2007. Incremental Text Structuring with Online Hierarchical Ranking. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Association for Computational Linguistics, Prague, Czech Republic, pages 83–91. <https://www.aclweb.org/anthology/D07-1009>. Cited on page 19.
- Hongbo Chen and Ben He. 2013. Automated essay scoring by maximizing human-machine agreement. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. pages 1741–1752. Cited on page 120.

- Mingda Chen, Zewei Chu, and Kevin Gimpel. 2019. Evaluation Benchmarks and Learning Criteria for Discourse-Aware Sentence Representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, pages 649–662. <https://www.aclweb.org/anthology/D19-1060>. Cited on page 101.
- Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent Attention Network on Memory for Aspect Sentiment Analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 452–461. <https://www.aclweb.org/anthology/D17-1047>. Cited on page 48.
- Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2016. Neural sentence ordering <https://arxiv.org/abs/1607.06952>. Cited on page 48.
- Jackie Chi Kit Cheung and Gerald Penn. 2010a. Entity-Based Local Coherence Modelling Using Topological Fields. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Uppsala, Sweden, pages 186–195. <https://www.aclweb.org/anthology/P10-1020>. Cited on pages 17 and 32.
- Jackie Chi Kit Cheung and Gerald Penn. 2010b. Utilizing Extra-Sentential Context for Parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Cambridge, MA, pages 23–33. <https://www.aclweb.org/anthology/D10-1003>. Cited on page 101.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1724–1734. <https://www.aclweb.org/anthology/D14-1179>. Cited on pages 34 and 39.
- François Chollet et al. 2015. Keras. <https://keras.io>. Cited on page 73.
- Noam Chomsky. 1957. *Syntactic Structures*. Mouton. Cited on page 103.
- Yu-An Chung, Hung-Yi Lee, and James Glass. 2018. Supervised and Unsupervised Transfer Learning for Question Answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New

Orleans, Louisiana, pages 1585–1594. <https://www.aclweb.org/anthology/N18-1143>.
Cited on pages 91 and 92.

Elizabeth Clark, Yangfeng Ji, and Noah A. Smith. 2018. Neural Text Generation in Stories Using Entity Representations as Context. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, pages 2250–2260. <https://www.aclweb.org/anthology/N18-1204>. Cited on page 20.

Kevin Clark and Christopher D. Manning. 2016. Deep Reinforcement Learning for Mention-Ranking Coreference Models. In *Empirical Methods on Natural Language Processing*. <https://nlp.stanford.edu/pubs/clark2016deep.pdf>. Cited on page 104.

Christelle Cocco, Raphaël Pittier, François Bavaud, and Aris Xanthos. 2011. Segmentation and Clustering of Textual Sequences: a Typological Approach. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*. Association for Computational Linguistics, Hissar, Bulgaria, pages 427–433. <https://www.aclweb.org/anthology/R11-1059>. Cited on page 102.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement* 20(1):37–46. Cited on page 49.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. pages 160–167. Cited on pages 35, 45, and 46.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.* 12:2493–2537. <http://dl.acm.org/citation.cfm?id=1953048.2078186>. Cited on pages 46, 47, and 68.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An Evaluation Toolkit for Universal Sentence Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), Miyazaki, Japan. <https://www.aclweb.org/anthology/L18-1269>. Cited on page 108.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark,

pages 670–680. <https://www.aclweb.org/anthology/D17-1070>. Cited on pages 42 and 111.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $\$&!#^*$ vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia. <https://www.aclweb.org/anthology/P18-1198>. Cited on pages 73 and 108.

Scott Crossley and Danielle McNamara. 2010. Cohesion, coherence, and expert evaluations of writing proficiency. In *Proceedings of the Annual Meeting of the Cognitive Science Society*. volume 32. Cited on page 129.

Scott Crossley and Danielle McNamara. 2011. Text coherence and judgments of essay quality: Models of quality and coherence. In *Proceedings of the Annual Meeting of the Cognitive Science Society*. volume 33. Cited on pages 18 and 129.

Scott A Crossley, David F Dufty, Philip M McCarthy, and Danielle S McNamara. 2007. Toward a new readability: A mixed model approach. In *Proceedings of the Annual Meeting of the Cognitive Science Society*. volume 29. Cited on page 19.

Scott A Crossley and Danielle S McNamara. 2016. Say More and Be More Coherent: How Text Elaboration and Cohesion Can Increase Writing Quality. *Grantee Submission* 7(3):351–370. Cited on page 24.

Baiyun Cui, Yingming Li, Ming Chen, and Zhongfei Zhang. 2018. Deep Attentive Sentence Ordering Network. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, pages 4340–4349. <https://www.aclweb.org/anthology/D18-1465>. Cited on pages 17, 19, 44, 45, and 99.

Baiyun Cui, Yingming Li, Yaqing Zhang, and Zhongfei Zhang. 2017. Text coherence analysis based on deep neural network. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. pages 2027–2030. Cited on pages 17 and 41.

Ronan Cummins and Marek Rei. 2018. Neural Multi-task Learning in Automated Assessment <https://arxiv.org/abs/1801.06830>. Cited on page 47.

Robert Dale. 1991a. Exploring the role of punctuation in the signalling of discourse structure. In *Proceedings of a workshop on text representation and domain modelling: ideas from linguistics and AI*. pages 110–120. Cited on page 84.

- Robert Dale. 1991b. The role of punctuation in discourse structure. In *Working Notes for the AAAI Fall Symposium on Discourse Structure in Natural Language Understanding and Generation*. pages 13–14. Cited on page 84.
- Frantisek Danes. 1974. Functional sentence perspective and the organization of the text. *Papers on functional sentence perspective* 106:128. Cited on page 16.
- Tirthankar Dasgupta, Abir Naskar, Lipika Dey, and Rupsa Saha. 2018. Augmenting textual qualitative features in deep convolution recurrent neural network for automatic essay scoring. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*. Association for Computational Linguistics, Melbourne, Australia, pages 93–102. <https://www.aclweb.org/anthology/W18-3713>. Cited on page 50.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 4171–4186. <https://www.aclweb.org/anthology/N19-1423>. Cited on pages 17, 37, 38, 39, and 106.
- Márcio de S Dias, Valéria D Feltrim, and Thiago Alexandre Salgueiro Pardo. 2014. Using rhetorical structure theory and entity grids to automatically evaluate local coherence in texts. In *International Conference on Computational Processing of the Portuguese Language*. Springer, pages 232–243. Cited on page 33.
- Tali Ditman and Gina R Kuperberg. 2010. Building coherence: A framework for exploring the breakdown of links across clause boundaries in schizophrenia. *Journal of neurolinguistics* 23(3):254–269. Cited on page 19.
- Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-Task Learning for Multiple Language Translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, pages 1723–1732. <https://www.aclweb.org/anthology/P15-1166>. Cited on page 46.
- Fei Dong and Yue Zhang. 2016. Automatic features for essay scoring—an empirical study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 1072–1077. Cited on pages 120 and 121.

- Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. pages 153–162. Cited on pages 50 and 121.
- Timothy Dozat and Christopher D. Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. <https://openreview.net/pdf?id=Hk95PK91e>. Cited on page 61.
- Amit Dubey, Patrick Sturt, and Frank Keller. 2005. Parallelism in Coordination as an Instance of Syntactic Priming: Evidence from Corpus-based Modeling. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Vancouver, British Columbia, Canada, pages 827–834. <https://www.aclweb.org/anthology/H05-1104>. Cited on page 101.
- Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. 2015. Low Resource Dependency Parsing: Cross-lingual Parameter Sharing in a Neural Network Parser. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Association for Computational Linguistics, Beijing, China, pages 845–850. <https://www.aclweb.org/anthology/P15-2139>. Cited on page 46.
- Nicholas D. Duran, Philip M. McCarthy, Art C. Graesser, and Danielle S. McNamara. 2007. Using temporal cohesion to predict temporal coherence in narrative and expository texts. *Behavior Research Methods* 39(2):212–223. Cited on page 24.
- Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent Neural Network Grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 199–209. <https://doi.org/10.18653/v1/N16-1024>. Cited on page 103.
- Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. HotFlip: White-Box Adversarial Examples for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Melbourne, Australia, pages 31–36. <https://www.aclweb.org/anthology/P18-2006>. Cited on page 48.
- Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14(2):179–211. Cited on page 33.

- Micha Elsner, Joseph Austerweil, and Eugene Charniak. 2007. A Unified Local and Global Model for Discourse Coherence. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*. Association for Computational Linguistics, Rochester, New York, pages 436–443. <https://www.aclweb.org/anthology/N07-1055>. Cited on pages 17 and 32.
- Micha Elsner and Eugene Charniak. 2008. Coreference-inspired Coherence Modeling. In *Proceedings of ACL-08: HLT, Short Papers*. Association for Computational Linguistics, pages 41–44. <https://www.aclweb.org/anthology/P08-2011>. Cited on pages 18, 19, 30, 32, and 68.
- Micha Elsner and Eugene Charniak. 2011a. Disentangling chat with local coherence models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. pages 1179–1189. Cited on page 20.
- Micha Elsner and Eugene Charniak. 2011b. Extending the Entity Grid with Entity-Specific Features. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pages 125–129. <https://www.aclweb.org/anthology/P11-2022>. Cited on pages 17, 19, 32, 43, 62, 68, 72, and 99.
- Brita Elvevåg, Peter W Foltz, Daniel R Weinberger, and Terry E Goldberg. 2007. Quantifying incoherence in speech: an automated methodology and novel application to schizophrenia. *Schizophrenia research* 93(1-3):304–316. Cited on pages 19 and 29.
- Gonenc Ercan and Ilyas Cicekli. 2008. Lexical cohesion based topic modeling for summarization. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, pages 582–592. Cited on page 29.
- Katrin Erk. 2012. Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass* 6(10):635–653. Cited on page 47.
- Allyson Ettinger, Ahmed Elgohary, Colin Phillips, and Philip Resnik. 2018. Assessing Composition in Sentence Vector Representations. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, pages 1790–1801. <https://www.aclweb.org/anthology/C18-1152>. Cited on page 108.
- Yimai Fang and Simone Teufel. 2014. A summariser based on human memory limitations and lexical competition. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*. pages 732–741. Cited on page 62.

- Younna Farag. 2016. *Convolutional Neural Networks for Automated Essay Assessment*. Master’s thesis, University of Cambridge, Computer Laboratory. <https://www.cl.cam.ac.uk/~yf273/mphil-dissertation.pdf>. Cited on page 47.
- Younna Farag, Marek Rei, and Ted Briscoe. 2017. An Error-Oriented Approach to Word Embedding Pre-Training. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, Copenhagen, Denmark, pages 149–158. <https://doi.org/10.18653/v1/W17-5016>. Cited on page 121.
- Younna Farag, Josef Valvoda, Helen Yannakoudakis, and Ted Briscoe. 2020. Analyzing neural discourse coherence models. In *Proceedings of the First Workshop on Computational Approaches to Discourse*. Association for Computational Linguistics, Online, pages 102–112. <https://www.aclweb.org/anthology/2020.codi-1.11>. Cited on page 100.
- Younna Farag and Helen Yannakoudakis. 2019. Multi-Task Learning for Coherence Modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, pages 629–639. <https://www.aclweb.org/anthology/P19-1060>. Cited on pages 67 and 99.
- Younna Farag, Helen Yannakoudakis, and Ted Briscoe. 2018. Neural Automated Essay Scoring and Coherence Modeling for Adversarially Crafted Input. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, pages 263–271. <https://www.aclweb.org/anthology/N18-1024/>. Cited on pages 49 and 119.
- Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting Word Vectors to Semantic Lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, pages 1606–1615. <https://www.aclweb.org/anthology/N15-1184>. Cited on page 47.
- Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of Neural Models Make Interpretations Difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, pages 3719–3728. <https://www.aclweb.org/anthology/D18-1407>. Cited on page 47.

- Vanessa Wei Feng and Graeme Hirst. 2012. Extending the Entity-based Coherence Model with Multiple Ranks. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Avignon, France, pages 315–324. <https://www.aclweb.org/anthology/E12-1032>. Cited on pages 17, 20, 81, and 145.
- Vanessa Wei Feng and Graeme Hirst. 2014. Patterns of local discourse coherence as a feature for authorship attribution. *Literary and Linguistic Computing* 29(2):191–198. Cited on page 20.
- Vanessa Wei Feng, Ziheng Lin, and Graeme Hirst. 2014. The Impact of Deep Hierarchical Discourse Structures in the Evaluation of Text Coherence. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, pages 940–949. <https://www.aclweb.org/anthology/C14-1089>. Cited on pages 17, 19, 33, 99, and 120.
- Elisa Ferracane, Su Wang, and Raymond Mooney. 2017. Leveraging discourse information effectively for authorship attribution. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. pages 584–593. Cited on page 20.
- Katja Filippova and Michael Strube. 2007. Extending the Entity-grid Coherence Model to Semantically Related Entities. In *Proceedings of the Eleventh European Workshop on Natural Language Generation (ENLG 07)*. DFKI GmbH, Saarbrücken, Germany, pages 139–142. <https://www.aclweb.org/anthology/W07-2321>. Cited on pages 17, 32, and 99.
- J. R. Firth. 1957. A synopsis of linguistic theory 1930-55. 1952-59:1–32. Cited on page 29.
- Peter W Foltz, Walter Kintsch, and Thomas K Landauer. 1998. The measurement of textual coherence with latent semantic analysis. *Discourse processes* 25(2-3):285–307. Cited on pages 16 and 29.
- Michel Galley, Kathleen R. McKeown, Eric Fosler-Lussier, and Hongyan Jing. 2003. Discourse Segmentation of Multi-Party Conversation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Sapporo, Japan, pages 562–569. <https://www.aclweb.org/anthology/P03-1071>. Cited on page 29.
- Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. 2018. Bottom-Up Abstractive Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, pages 4098–4109. <https://www.aclweb.org/anthology/D18-1443>. Cited on page 37.

- Morton Ann Gernsbacher and David J Hargreaves. 1988. Accessing sentence participants: The advantage of first mention. *Journal of Memory and Language* 27(6):699–717. <https://www.sciencedirect.com/science/article/pii/0749596X88900162>. Cited on page 27.
- Rachel Giora. 1985. Notes towards a theory of text coherence. *Poetics today* 6(4):699–715. Cited on page 24.
- T Givón. 1995. Coherence in text vs. coherence in mind. *Coherence in spontaneous text* 31:59. Cited on page 16.
- T. Givón. 1992. The grammar of referential coherence as mental processing instructions. *Linguistics* 30(1):5 – 56. <https://www.degruyter.com/view/journals/ling/30/1/article-p5.xml>. Cited on page 27.
- Malcolm Gladwell. 2017. *Outliers: the story of success*. Audio-Tech Business Book Summaries. Cited on page 15.
- Goran Glavaš and Swapna Somasundaran. 2020. Two-Level Transformer and Auxiliary Coherence Modeling for Improved Text Segmentation. In *AAAI Publications, Thirty-Four AAAI Conference on Artificial Intelligence*. AAAI Press. Cited on page 20.
- Yoav Goldberg. 2019. Assessing BERT’s Syntactic Abilities. <https://arxiv.org/abs/1901.05287>. Cited on page 103.
- Jingjing Gong, Xinchu Chen, Xipeng Qiu, and Xuanjing Huang. 2016. End-to-end neural sentence ordering using pointer network. *arXiv preprint arXiv:1611.04953* . Cited on pages 19 and 44.
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>. Cited on pages 35 and 71.
- Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. <http://arxiv.org/abs/1412.6572>. Cited on page 48.
- Peter C Gordon, Barbara J Grosz, and Laura A Gilliom. 1993. Pronouns, names, and the centering of attention in discourse. *Cognitive science* 17(3):311–347. Cited on pages 16 and 27.
- Stephan Gouws, Yoshua Bengio, and Greg Corrado. 2015. BilBOWA: Fast Bilingual Distributed Representations without Word Alignments. In *Proceedings of the 32nd*

- International Conference on Machine Learning*. PMLR, Lille, France, volume 37 of *Proceedings of Machine Learning Research*, pages 748–756. <http://proceedings.mlr.press/v37/gouws15.html>. Cited on page 37.
- Arthur C Graesser, Danielle S McNamara, Max M Louwerse, and Zhiqiang Cai. 2004. Coh-Metrix: Analysis of text on cohesion and language. *Behavior research methods, instruments, & computers* 36(2):193–202. Cited on page 19.
- Arthur C Graesser, Murray Singer, and Tom Trabasso. 1994. Constructing inferences during narrative text comprehension. *Psychological review* 101(3):371. Cited on pages 16 and 25.
- Herbert P Grice. 1975. Logic and conversation. In *Speech acts*, Brill, pages 41–58. Cited on page 25.
- Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1983. PROVIDING A UNIFIED ACCOUNT OF DEFINITE NOUN PHRASES IN DISCOURSE. In *21st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Cambridge, Massachusetts, USA, pages 44–50. <https://www.aclweb.org/anthology/P83-1007>. Cited on page 27.
- Barbara J. Grosz and Candace L. Sidner. 1986. Attention, Intentions, and the Structure of Discourse. *Computational Linguistics* 12(3):175–204. <https://www.aclweb.org/anthology/J86-3001>. Cited on pages 16, 17, and 26.
- Barbara J. Grosz, Scott Weinstein, and Aravind K. Joshi. 1995. Centering: A Framework for Modeling the Local Coherence of Discourse. *Computational Linguistics* 21(2). <https://www.aclweb.org/anthology/J95-2003>. Cited on pages 16, 17, 18, 27, 28, and 82.
- Camille Guinaudeau and Michael Strube. 2013. Graph-based Local Coherence Modeling. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 93–103. <https://www.aclweb.org/anthology/P13-1010>. Cited on pages 17, 19, 32, 33, 45, 62, and 99.
- Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless Green Recurrent Networks Dream Hierarchically. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, pages 1195–1205. <https://www.aclweb.org/anthology/N18-1108>. Cited on pages 102 and 110.

- Jeanette K. Gundel, Nancy Hedberg, and Ron Zacharski. 1993. Cognitive Status and the Form of Referring Expressions in Discourse. *Language* 69:274. <https://www.jstor.org/stable/pdf/416535.pdf>. Cited on page 27.
- Karl Haberlandt. 1982. Reader Expectations in Text Comprehension. In Jean-François [Le Ny] and Walter Kintsch, editors, *Language and Comprehension*, North-Holland, volume 9 of *Advances in Psychology*, pages 239 – 249. <http://www.sciencedirect.com/science/article/pii/S0166411509600558>. Cited on page 24.
- Sherzod Hakimov, Hakan Tunc, Marlen Akimaliev, and Erdogan Dogdu. 2013. Semantic Question Answering System over Linked Data Using Relational Patterns. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*. ACM, New York, NY, USA, EDBT '13, pages 83–88. <http://doi.acm.org/10.1145/2457317.2457331>. Cited on page 62.
- M.A.K. Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. Longman, Dallas, Texas. Cited on pages 15, 16, 23, and 24.
- Christian Hardmeier. 2014. *Discourse in Statistical Machine Translation*. Ph.D. thesis, Uppsala University, Department of Linguistics and Philology. Cited on page 20.
- Zellig S Harris. 1954. Distributional structure. *Word* 10(2-3):146–162. Cited on page 29.
- Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 1923–1933. <https://www.aclweb.org/anthology/D17-1206>. Cited on pages 46 and 47.
- Laura Hasler. 2004. An investigation into the use of centering transitions for summarisation. In *Proceedings of the 7th Annual CLUK Research Colloquium*. pages 100–107. Cited on page 17.
- Susan E. Haviland and Herbert H. Clark. 1974. What’s new? Acquiring New information as a process in comprehension. *Journal of Verbal Learning and Verbal Behavior* 13(5):512 – 521. <http://www.sciencedirect.com/science/article/pii/S0022537174800034>. Cited on page 24.
- John Hewitt and Christopher D. Manning. 2019. A Structural Probe for Finding Syntax in Word Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 4129–4138. <https://www.aclweb.org/anthology/N19-1419>. Cited on pages 38, 56, and 108.

- D. Higgins, J. Burstein, and Y. Attali. 2006. Identifying Off-Topic Student Essays without Topic-Specific Training Data. *Nat. Lang. Eng.* 12(2):145–159. <https://doi.org/10.1017/S1351324906004189>. Cited on page 122.
- Derrick Higgins and Jill Burstein. 2007. Sentence similarity measures for essay coherence. In *Proceedings of the 7th International Workshop on Computational Semantics*. pages 1–12. https://www.ets.org/Media/Research/pdf/erater_sentence_similarity.pdf. Cited on pages 19, 29, and 120.
- Derrick Higgins, Jill Burstein, Daniel Marcu, and Claudia Gentile. 2004. Evaluating Multiple Aspects of Coherence in Student Essays. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*. Association for Computational Linguistics, Boston, Massachusetts, USA, pages 185–192. <https://www.aclweb.org/anthology/N04-1024>. Cited on pages 16, 18, 29, and 120.
- Derrick Higgins and Michael Heilman. 2014. Managing What We Can Measure: Quantifying the Susceptibility of Automated Scoring Systems to Gaming Behavior. *Educational Measurement: Issues and Practice* 33:36–46. <https://onlinelibrary.wiley.com/doi/pdf/10.1111/emip.12036>. Cited on page 122.
- Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation* 14(8):1771–1800. Cited on page 130.
- Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors <https://arxiv.org/pdf/1207.0580.pdf>. Cited on page 73.
- Graeme Hirst, David St-Onge, et al. 1998. Lexical chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database* 305:305–332. Cited on page 29.
- Jerry R Hobbs. 1979. Coherence and coreference. *Cognitive science* 3(1):67–90. Cited on pages 16, 24, and 25.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* 9(8):1735–1780. Cited on page 34.
- Michael Hoey. 2005. *Lexical priming: A new theory of words and language*. Psychology Press. Cited on page 16.
- Fabian Hommel, Philipp Cimiano, Matthias Orlikowski, and Matthias Hartung. 2019. Extending Neural Question Answering with Linguistic Input Features. In *Proceedings of*

- the 5th Workshop on Semantic Deep Learning (SemDeep-5)*. Association for Computational Linguistics, Macau, China, pages 31–39. <https://www.aclweb.org/anthology/W19-5806>. Cited on page 64.
- Matthew Honnibal and Mark Johnson. 2015. An Improved Non-monotonic Transition System for Dependency Parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 1373–1378. <https://aclweb.org/anthology/D/D15/D15-1162>. Cited on pages 70 and 104.
- Hossein Hosseini, Sreeram Kannan, Baosen Zhang, and Radha Poovendran. 2017. Deceiving google’s perspective api built for detecting toxic comments <https://arxiv.org/abs/1702.08138>. Cited on page 48.
- Eduard H. Hovy. 1990. Parsimonious and Profligate Approaches to the Question of Discourse Structure Relations. In *Proceedings of the Fifth International Workshop on Natural Language Generation*. <https://www.aclweb.org/anthology/W90-0117>. Cited on page 25.
- Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, pages 328–339. <https://www.aclweb.org/anthology/P18-1031>. Cited on pages 37, 92, and 146.
- Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, pages 873–882. Cited on page 36.
- Guimin Huang, Min Tan, Zhenglin Sun, and Ya Zhou. 2018. RST-based Discourse Coherence Quality Analysis Model for Students’ English Essays. In *MATEC Web of Conferences*. EDP Sciences, volume 232, page 02020. Cited on pages 33 and 120.
- Dan Iter, Jong Yoon, and Dan Jurafsky. 2018. Automatic detection of incoherent speech for diagnosing schizophrenia. In *Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic*. pages 136–146. Cited on pages 15, 19, and 29.
- Ganesh Jawahar, Benoît Sagot, and Djamel Seddah. 2019. What Does BERT Learn about the Structure of Language? In *Proceedings of the 57th Annual Meeting of the Association*

- for *Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, pages 3651–3657. <https://www.aclweb.org/anthology/P19-1356>. Cited on pages 38 and 56.
- Yacine Jernite, Samuel R Bowman, and David Sontag. 2017. Discourse-based objectives for fast unsupervised sentence representation learning <https://arxiv.org/abs/1705.00557>. Cited on pages 47 and 145.
- Robin Jia and Percy Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 2021–2031. <https://www.aclweb.org/anthology/D17-1215>. Cited on page 48.
- Cancan Jin, Ben He, Kai Hui, and Le Sun. 2018. TDNN: a two-stage deep neural network for prompt-independent automated essay scoring. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pages 1088–1097. Cited on page 50.
- Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics* 5:339–351. Cited on page 47.
- Aravind K Joshi and Scott Weinstein. 1981. Control of Inference: Role of Some Aspects of Discourse Structure-Centering. In *IJCAI*. pages 385–387. Cited on pages 16 and 27.
- Shafiq Joty, Francisco Guzmán, Lluís Màrquez, and Preslav Nakov. 2017. Discourse structure in machine translation evaluation. *Computational Linguistics* 43(4):683–722. Cited on page 20.
- Shafiq Joty, Muhammad Tasnim Mohiuddin, and Dat Tien Nguyen. 2018. Coherence Modeling of Asynchronous Conversations: A Neural Entity Grid Approach. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, pages 558–568. <https://www.aclweb.org/anthology/P18-1052>. Cited on pages 17, 20, 43, 62, 72, and 99.
- Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification <http://arxiv.org/abs/1607.01759>. Cited on page 73.

- Ákos Kádár, Grzegorz Chrupała, and Afra Alishahi. 2017. Representation of Linguistic Form and Function in Recurrent Neural Networks. *Computational Linguistics* 43(4):761–780. <https://www.aclweb.org/anthology/J17-4003>. Cited on page 48.
- Megumi Kameyama. 1998. Intrasentential Centering: A Case Study. In M. Walker, A. Joshi, and E. Prince (eds.), editors, *Centering Theory in Discourse*, Clarendon Press, Oxford, chapter 6, page 89–112. Cited on pages 27 and 82.
- Nikiforos Karamanis. 2001. Exploring entity-based coherence. In *University of Sheffield. Citeseer*. Cited on page 17.
- Nikiforos Karamanis, Massimo Poesio, Chris Mellish, and Jon Oberlander. 2004. Evaluating centering-based metrics of coherence for text structuring using a reliably annotated corpus. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, page 391. Cited on page 17.
- Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2020. End-to-end bias mitigation by modelling biases in corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, pages 8706–8716. <https://www.aclweb.org/anthology/2020.acl-main.769>. Cited on page 130.
- Andrej Karpathy, Justin Johnson, and Fei-Fei Li. 2016. Visualizing and Understanding Recurrent Networks. In *International Conference on Learning Representations (ICLR) Workshop*. San Juan, Puerto Rico. <http://arxiv.org/abs/1506.02078>. Cited on page 48.
- Andrew Kehler and Andrew Kehler. 2002. *Coherence, reference, and the theory of grammar*. CSLI publications Stanford, CA. Cited on page 25.
- Maurice G Kendall and J Dickinson Gibbons. 1990. *Rank correlation methods*. Oxford University Press, New York. Cited on page 50.
- Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1746–1751. <https://doi.org/10.3115/v1/D14-1181>. Cited on page 35.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. <http://arxiv.org/abs/1412.6980>. Cited on page 108.

- Walter Kintsch and Teun A Van Dijk. 1978. Toward a model of text comprehension and production. *Psychological review* 85(5):363. Cited on pages 16 and 62.
- Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-Thought Vectors. In *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., pages 3294–3302. <http://papers.nips.cc/paper/5950-skip-thought-vectors.pdf>. Cited on page 40.
- Fajri Koto, Jey Han Lau, and Timothy Baldwin. 2019. Improved Document Modelling with a Neural Discourse Parser. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*. Australasian Language Technology Association, Sydney, Australia, pages 67–76. <https://www.aclweb.org/anthology/U19-1010>. Cited on page 20.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, Curran Associates, Inc., pages 1097–1105. <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>. Cited on page 75.
- Pawan Kumar, Dhanajit Brahma, Harish Karnick, and Piyush Rai. 2020a. Deep Attentive Ranking Networks for Learning to Order Sentences. In *AAAI Publications, Thirty-Four AAAI Conference on Artificial Intelligence*. AAAI Press. Cited on pages 17, 20, and 45.
- Yaman Kumar, Mehar Bhatia, Anubha Kabra, Jessy Junyi Li, Di Jin, and Rajiv Ratn Shah. 2020b. Calling Out Bluff: Attacking the Robustness of Automatic Scoring Systems with Simple Adversarial Testing. <https://arxiv.org/abs/2007.06796>. Cited on pages 122 and 146.
- Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modeling Structure Makes Them Better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, pages 1426–1436. <https://www.aclweb.org/anthology/P18-1132>. Cited on pages 102, 103, and 110.
- Alice Lai and Joel Tetreault. 2018. Discourse Coherence in the Wild: A Dataset, Evaluation and Methods. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*. Association for Computational Linguistics, Melbourne, Australia, pages 214–223. <https://www.aclweb.org/anthology/W18-5023>. Cited on pages 18, 19, 69, 70, 71, 73, 75, 88, 89, 92, 93, and 99.

- Thomas K Landauer. 2003. Automated scoring and annotation of essays with the Intelligent Essay Assessor. *Automated essay scoring: A cross-disciplinary perspective* <https://ci.nii.ac.jp/naid/10018295871/en/>. Cited on pages 29 and 120.
- Thomas K Landauer and Susan T Dumais. 1997. A solution to Plato’s problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review* 104(2):211. Cited on page 28.
- J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33(1):159–174. <http://www.jstor.org/stable/2529310>. Cited on pages 69 and 107.
- Mirella Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 545–552. Cited on pages 19, 20, 30, 62, and 64.
- Mirella Lapata. 2006. Automatic Evaluation of Information Ordering: Kendall’s Tau. *Comput. Linguist.* 32(4):471–484. Cited on pages 50 and 81.
- Mirella Lapata and Regina Barzilay. 2005. Automatic Evaluation of Text Coherence: Models and Representations. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, IJCAI’05, pages 1085–1090. <http://dl.acm.org/citation.cfm?id=1642293.1642467>. Cited on pages 29 and 99.
- Alex Lascarides and Nicholas Asher. 1991. DISCOURSE RELATIONS AND DEFEASIBLE KNOWLEDGE ‘. In *29th Annual Meeting of the Association for Computational Linguistics*. pages 55–62. Cited on page 16.
- Alex Lascarides and Nicholas Asher. 1993. Temporal interpretation, discourse relations and commonsense entailment. *Linguistics and philosophy* 16(5):437–493. Cited on page 25.
- Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*. JMLR.org, ICML’14, page II–1188–II–1196. Cited on page 40.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324. Cited on page 35.

- Stephen C. Levinson. 1983. *Pragmatics*. Cambridge Textbooks in Linguistics. Cambridge University Press. Cited on pages 15 and 25.
- Omer Levy and Yoav Goldberg. 2014. Linguistic Regularities in Sparse and Explicit Word Representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Ann Arbor, Michigan, pages 171–180. <https://www.aclweb.org/anthology/W14-1618>. Cited on page 47.
- Jing Li, Le Sun, Chunyu Kit, and Jonathan Webster. 2007. A query-focused multi-document summarizer based on lexical chains. In *Proceedings of the Document Understanding Conference DUC-2007*. <https://pdfs.semanticscholar.org/40e7/eacca5108e9368116c7356e96e5ddb72f930.pdf>. Cited on page 29.
- Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016a. Visualizing and Understanding Neural Models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 681–691. <https://www.aclweb.org/anthology/N16-1082>. Cited on page 48.
- Jiwei Li and Eduard Hovy. 2014. A Model of Coherence Based on Distributed Sentence Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 2039–2048. <https://doi.org/10.3115/v1/D14-1218>. Cited on pages 17, 19, 40, 70, 71, and 75.
- Jiwei Li and Dan Jurafsky. 2017. Neural Net Models of Open-domain Discourse Coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 198–209. <https://doi.org/10.18653/v1/D17-1019>. Cited on pages 17, 19, 40, 41, 43, 70, 71, 75, 99, and 100.
- Jiwei Li, Thang Luong, and Dan Jurafsky. 2015. A Hierarchical Neural Autoencoder for Paragraphs and Documents. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, pages 1106–1115. <https://doi.org/10.3115/v1/P15-1107>. Cited on pages 40 and 53.
- Jiwei Li, Will Monroe, and Dan Jurafsky. 2016b. Understanding neural networks through representation erasure <https://arxiv.org/abs/1612.08220>. Cited on page 48.

- X. Li, Q. Wen, and K. Pan. 2017. Unsupervised Off-Topic Essay Detection Based on Target and Reference Prompts. In *2017 13th International Conference on Computational Intelligence and Security (CIS)*. pages 465–468. Cited on page 122.
- Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-Attentive Sentence Embedding. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. https://openreview.net/forum?id=BJC_jUqxe. Cited on pages 39 and 48.
- Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing Implicit Discourse Relations in the Penn Discourse Treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Singapore, pages 343–351. <https://www.aclweb.org/anthology/D09-1036>. Cited on page 64.
- Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2011. Automatically Evaluating Text Coherence Using Discourse Relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, pages 997–1006. <https://www.aclweb.org/anthology/P11-1100>. Cited on pages 17, 33, 68, and 99.
- Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics* 4:521–535. <https://www.aclweb.org/anthology/Q16-1037>. Cited on pages 102 and 110.
- C. Liu, W. Hsaio, C. Lee, and H. Chi. 2013. An HMM-Based Algorithm for Content Ranking and Coherence-Feature Extraction. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 43(2):440–450. <https://ieeexplore.ieee.org/document/6408207>. Cited on page 120.
- Jiawei Liu, Yang Xu, and Lingzhe Zhao. 2019a. Automated essay scoring based on two-stage learning. *arXiv preprint arXiv:1901.07744* . Cited on page 121.
- Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019b. Linguistic Knowledge and Transferability of Contextual Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, pages 1073–1094. <https://doi.org/10.18653/v1/N19-1112>. Cited on pages 39, 56, and 108.

- Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent Neural Network for Text Classification with Multi-Task Learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press, IJCAI'16, page 2873–2879. Cited on page 46.
- Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019c. Multi-Task Deep Neural Networks for Natural Language Understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, pages 4487–4496. <https://www.aclweb.org/anthology/P19-1441>. Cited on page 39.
- Yang Liu and Mirella Lapata. 2019. Text Summarization with Pretrained Encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, pages 3730–3740. <https://www.aclweb.org/anthology/D19-1387>. Cited on page 37.
- Karen E Lochbaum, Mark Rosenstein, PW Foltz, Marcia A Derr, et al. 2013. Detection of gaming in automated scoring of essays with the IEA. *Presented at the 75th Annual meeting of NCME* . Cited on page 121.
- Lajanugen Logeswaran, Honglak Lee, and Dragomir R. Radev. 2018. Sentence Ordering and Coherence Modeling using Recurrent Neural Networks. In *AAAI Publications, Thirty-Second AAAI Conference on Artificial Intelligence*. AAAI Press, pages 5285–5292. <https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17011/16079>. Cited on pages 17, 19, 44, 48, and 99.
- Annie Louis and Derrick Higgins. 2010. Off-topic essay detection using short prompt texts. In *proceedings of the NAACL HLT 2010 fifth workshop on innovative use of NLP for building educational applications*. Association for Computational Linguistics, pages 92–95. <https://www.aclweb.org/anthology/W10-1013.pdf>. Cited on page 122.
- Annie Louis and Ani Nenkova. 2012. A Coherence Model Based on Syntactic Patterns. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, Jeju Island, Korea, pages 1157–1168. <https://www.aclweb.org/anthology/D12-1106>. Cited on pages 17, 18, 30, 64, 99, 100, and 101.
- Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., pages 4765–4774. <http://papers.nips.cc/paper/>

7062-a-unified-approach-to-interpreting-model-predictions.pdf. Cited on page 47.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Bilingual Word Representations with Monolingual Quality in Mind. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. Association for Computational Linguistics, Denver, Colorado, pages 151–159. <https://doi.org/10.3115/v1/W15-1521>. Cited on page 37.

Htet Myet Lynn, Chang Choi, and Pankoo Kim. 2018. An improved method of automatic text summarization for web contents using lexical chain with semantic-related terms. *Soft Computing* 22(12):4013–4023. Cited on page 29.

Okumura Manabu and Honda Takeo. 1994. Word sense disambiguation and text segmentation based on lexical cohesion. In *Proceedings of the 15th conference on Computational linguistics- Volume 2*. Association for Computational Linguistics, pages 755–761. Cited on page 29.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text Interdisciplinary Journal for the Study of Discourse* pages 243–281. Cited on pages 16, 17, 25, 26, and 99.

Meghana Marathe and Graeme Hirst. 2010. Lexical chains using distributional measures of concept distance. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, pages 291–302. Cited on page 29.

Daniel Marcu. 1997. From local to global coherence: A bottom-up approach to text planning. In *AAAI/IAAI*. pages 629–635. Cited on page 16.

Rebecca Marvin and Tal Linzen. 2018. Targeted Syntactic Evaluation of Language Models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, pages 1192–1202. <https://www.aclweb.org/anthology/D18-1151>. Cited on page 103.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in Translation: Contextualized Word Vectors. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, Curran Associates, Inc., pages 6294–6305. <http://papers.nips.cc/paper/7209-learned-in-translation-contextualized-word-vectors.pdf>. Cited on page 37.

- George A McCulley. 1985. Writing quality, coherence, and cohesion. *Research in the Teaching of English* pages 269–282. Cited on page 24.
- Neil McIntyre and Mirella Lapata. 2010. Plot induction and evolutionary search for story generation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 1562–1572. Cited on page 20.
- Danielle S McNamara. 2001. Reading both high-coherence and low-coherence texts: Effects of text sequence and prior knowledge. *Canadian Journal of Experimental Psychology/Revue canadienne de psychologie expérimentale* 55(1):51. Cited on page 24.
- Mohsen Mesgar and Michael Strube. 2015. Graph-based Coherence Modeling For Assessing Readability. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics, Denver, Colorado, pages 309–318. <https://doi.org/10.18653/v1/S15-1036>. Cited on pages 17, 19, and 33.
- Mohsen Mesgar and Michael Strube. 2016. Lexical Coherence Graph Modeling Using Word Embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 1414–1423. <https://doi.org/10.18653/v1/N16-1167>. Cited on page 19.
- Mohsen Mesgar and Michael Strube. 2018. A Neural Local Coherence Model for Text Quality Assessment. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, pages 4328–4339. <https://www.aclweb.org/anthology/D18-1464>. Cited on pages 17, 41, and 121.
- Thomas Meyer, Andrei Popescu-Belis, Najeh Hajlaoui, and Andrea Gesmundo. 2012. Machine translation of labeled discourse connectives. In *Proceedings of the Tenth Biennial Conference of the Association for Machine Translation in the Americas (AMTA)*. CONF. Cited on page 20.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* . Cited on page 36.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. <http://arxiv.org/abs/1301.3781>. Cited on page 47.

- Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhersch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), Miyazaki, Japan. <https://www.aclweb.org/anthology/L18-1008>. Cited on pages 17, 37, and 73.
- Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2011. Extensions of recurrent neural network language model. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, pages 5528–5531. Cited on page 36.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013c. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. pages 3111–3119. Cited on pages 17, 37, 47, and 72.
- Eleni Miltsakaki and Karen Kukich. 2000. Automated evaluation of coherence in student essays. In *Proceedings of LREC 2000*. pages 1–8. Cited on pages 17, 19, 31, and 120.
- Farjana Sultana Mim, Naoya Inoue, Paul Reisert, Hiroki Ouchi, and Kentaro Inui. 2019. Unsupervised Learning of Discourse-Aware Text Representation for Essay Scoring. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. pages 378–385. Cited on page 121.
- Sewon Min, Minjoon Seo, and Hannaneh Hajishirzi. 2017. Question Answering through Transfer Learning from Large Fine-grained Supervision Data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 510–517. <https://www.aclweb.org/anthology/P17-2081>. Cited on page 91.
- Jeff Mitchell and Mirella Lapata. 2008. Vector-based Models of Semantic Composition. In *Proceedings of ACL-08: HLT*. Association for Computational Linguistics, Columbus, Ohio, pages 236–244. <https://www.aclweb.org/anthology/P08-1028>. Cited on page 39.
- Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in neural information processing systems*. pages 2265–2273. Cited on page 37.

- Han Cheol Moon, Tasnim Mohiuddin, Shafiq Joty, and Chi Xu. 2019. A Unified Neural Coherence Model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, pages 2262–2272. <https://doi.org/10.18653/v1/D19-1231>. Cited on pages 41, 82, 99, and 145.
- Jane Morris and Graeme Hirst. 1991. Lexical Cohesion Computed by Thesaural relations as an indicator of the structure of text. *Computational Linguistics* 17(1):21–48. <https://www.aclweb.org/anthology/J91-1002>. Cited on pages 15, 16, 28, and 29.
- Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 839–849. <https://doi.org/10.18653/v1/N16-1098>. Cited on pages 103, 104, and 105.
- Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How Transferable are Neural Networks in NLP Applications? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 479–489. <https://www.aclweb.org/anthology/D16-1046>. Cited on page 91.
- Pramod Kaushik Mudrakarta, Ankur Taly, Mukund Sundararajan, and Kedar Dhamdhere. 2018. Did the Model Understand the Question? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, pages 1896–1906. <https://www.aclweb.org/anthology/P18-1176>. Cited on page 48.
- Farah Nadeem, Huy Nguyen, Yang Liu, and Mari Ostendorf. 2019. Automated Essay Scoring with Discourse-Aware Neural Models. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, Florence, Italy, pages 484–493. <https://doi.org/10.18653/v1/W19-4450>. Cited on pages 19, 121, 126, 127, 132, and 134.
- Farah Nadeem and Mari Ostendorf. 2018. Estimating Linguistic Complexity for Science Texts. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, New Orleans, Louisiana, pages 45–55. <https://doi.org/10.18653/v1/W18-0505>. Cited on page 126.

- Jan Niehues and Eunah Cho. 2017. Exploiting Linguistic Resources for Neural Machine Translation Using Multi-task Learning. In *Proceedings of the Second Conference on Machine Translation*. Association for Computational Linguistics, Copenhagen, Denmark, pages 80–89. <https://www.aclweb.org/anthology/W17-4708>. Cited on page 64.
- Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), Portorož, Slovenia, pages 1659–1666. <https://www.aclweb.org/anthology/L16-1262>. Cited on page 62.
- Byungkook Oh, Seungmin Seo, Cheolheon Shin, Eunju Jo, and Kyong-Ho Lee. 2019. Topic-Guided Coherence Modeling for Sentence Ordering by Preserving Global and Local Information. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, pages 2273–2283. <https://www.aclweb.org/anthology/D19-1232>. Cited on pages 20, 44, and 99.
- Ellis B Page. 1968. The use of the computer in analyzing student essays. *International review of education* pages 210–225. Cited on page 120.
- D. Palma and J. Atkinson. 2018. Coherence-Based Automatic Essay Assessment. *IEEE Intelligent Systems* 33(5):26–36. Cited on pages 19, 29, and 120.
- Daraksha Parveen and Michael Strube. 2015. Integrating importance, non-redundancy and coherence in graph-based extractive summarization. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*. Cited on page 20.
- Anayeli Paulino, Gerardo Sierra, Laura Hernández-Domínguez, Iria da Cunha, and Gemma Bel-Enguix. 2018. Rhetorical relations in the speech of Alzheimer’s patients and healthy elderly subjects: An approach from the RST. *Computación y Sistemas* 22(3):895–905. Cited on page 19.
- Ellie Pavlick and Joel Tetreault. 2016. An empirical analysis of formality in online communication. *Transactions of the Association for Computational Linguistics* 4:61–74. <https://www.aclweb.org/anthology/Q16-1005>. Cited on page 69.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods*

- in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1532–1543. <https://www.aclweb.org/anthology/D14-1162>. Cited on pages 17 and 37.
- Isaac Persing and Vincent Ng. 2014. Modeling prompt adherence in student essays. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. pages 1534–1543. <https://www.aclweb.org/anthology/P14-1144.pdf>. Cited on page 122.
- Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 1756–1765. <https://www.aclweb.org/anthology/P17-1161>. Cited on pages 38, 56, and 135.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, pages 2227–2237. <https://www.aclweb.org/anthology/N18-1202>. Cited on pages 17, 37, 38, 56, and 135.
- Casper Petersen, Christina Lioma, Jakob Grue Simonsen, and Birger Larsen. 2015. Entropy and graph based modelling of document coherence using discourse entities: An application to IR. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*. pages 191–200. Cited on page 20.
- Peter Phandi, Kian Ming A. Chai, and Hwee Tou Ng. 2015. Flexible Domain Adaptation for Automated Essay Scoring Using Correlated Linear Regression. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 431–439. <https://doi.org/10.18653/v1/D15-1049>. Cited on pages 50 and 120.
- Christian Pietsch, Armin Buch, Stefan Kopp, and Jan de Ruiter. 2012. Measuring syntactic priming in dialogue corpora. *Empirical approaches to linguistic theory: Studies in meaning and structure* 111:29. Cited on page 101.
- Emily Pitler, Annie Louis, and Ani Nenkova. 2010. Automatic Evaluation of Linguistic Quality in Multi-Document Summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Uppsala, Sweden, pages 544–554. <https://www.aclweb.org/anthology/P10-1056>. Cited on page 20.

- Emily Pitler and Ani Nenkova. 2008. Revisiting Readability: A Unified Framework for Predicting Text Quality. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Honolulu, Hawaii, pages 186–195. <https://www.aclweb.org/anthology/D08-1020>. Cited on page 19.
- Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 412–418. <https://www.aclweb.org/anthology/P16-2067>. Cited on page 46.
- Donald E. Powers, Jill Burstein, Martin Chodorow, Mary E. Fowles, and Karen Kukich. 2002. Stumping e-rater: challenging the validity of automated essay scoring. *Computers in Human Behavior* 18(2):103–134. Cited on pages 121 and 122.
- Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08)*. European Language Resources Association (ELRA), Marrakech, Morocco. Cited on page 26.
- Ellen Prince. 1981. Toward a taxonomy of given - new information. In Peter Cole, editor, *Radical pragmatics*, Academic Press, New York, pages 223–255. Cited on page 27.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language_understanding_paper.pdf. Cited on page 37.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1(8):9. Cited on page 37.
- Zahra Rahimi, Diane Litman, Elaine Wang, and Richard Correnti. 2015. Incorporating Coherence of Topics as a Criterion in Automatic Response-to-Text Assessment of the Organization of Writing. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, Denver, Colorado, pages 20–30. <https://www.aclweb.org/anthology/W15-0603>. Cited on pages 29 and 120.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don’t Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of*

- the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Melbourne, Australia, pages 784–789. <https://www.aclweb.org/anthology/P18-2124>. Cited on page 37.
- Owen Rambow. 1993. Pragmatic Aspects of Scrambling and Topicalization in German: A Centering Approach. In *University of Pennsylvania*. Cited on page 27.
- Gisela Redeker. 1990. Ideational and pragmatic markers of discourse structure. *Journal of Pragmatics* 14(3):367 – 381. Special Issue: 'Selected papers from The International Pragmatics Conference, Antwerp, 17-22 August, 1987'. [https://doi.org/https://doi.org/10.1016/0378-2166\(90\)90095-U](https://doi.org/https://doi.org/10.1016/0378-2166(90)90095-U). Cited on pages 15, 16, and 25.
- Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, Valletta, Malta, pages 45–50. <http://is.muni.cz/publication/884893/en>. Cited on page 77.
- Marek Rei. 2017. Semi-supervised Multitask Learning for Sequence Labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 2121–2130. <https://doi.org/10.18653/v1/P17-1194>. Cited on page 46.
- Marek Rei and Ronan Cummins. 2016. Sentence Similarity Measures for Fine-Grained Estimation of Topical Relevance in Learner Essays. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, San Diego, CA, pages 283–288. <https://www.aclweb.org/anthology/W16-0533>. Cited on page 122.
- Marek Rei and Anders Søgaard. 2019. Jointly learning to label sentences and tokens. In *Proceedings of the AAAI Conference on Artificial Intelligence*. volume 33, pages 6916–6923. Cited on page 47.
- Marek Rei and Helen Yannakoudakis. 2017. Auxiliary Objectives for Neural Error Detection Models. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, Copenhagen, Denmark, pages 33–43. <https://www.aclweb.org/anthology/W17-5004>. Cited on page 46.
- Nils Reimers and Iryna Gurevych. 2019. Alternative weighting schemes for elmo embeddings. *arXiv preprint arXiv:1904.02954* . Cited on page 135.
- David Reitter, Johanna D Moore, and Frank Keller. 2006. Priming of syntactic rules in task-oriented dialogue and spontaneous conversation . Cited on page 101.

- Steffen Remus and Chris Biemann. 2013. Three Knowledge-Free Methods for Automatic Lexical Chain Extraction. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Atlanta, Georgia, pages 989–999. <https://www.aclweb.org/anthology/N13-1119>. Cited on page 29.
- Brian Riordan, Andrea Horbach, Aoife Cahill, Torsten Zesch, and Chungmin Lee. 2017. Investigating neural architectures for short answer scoring. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. pages 159–168. Cited on page 121.
- Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* . Cited on pages 18, 46, and 146.
- Lawrence M. Rudner and Tahung Liang. 2002. Automated Essay Scoring Using Bayes’ Theorem. *The Journal of Technology, Learning and Assessment* 1(2). <https://ejournals.bc.edu/index.php/jtla/article/view/1668>. Cited on page 120.
- LM Rudner, Veronica Garcia, and Catherine Welch. 2006. An evaluation of IntelliMetric essay scoring system. *The Journal of Technology, Learning, and Assessment* 4(4):1 – 22. <https://ejournals.bc.edu/index.php/jtla/article/view/1651>. Cited on page 120.
- Vasile Rus and Nobal Niraula. 2012a. Automated Detection of Local Coherence in Short Argumentative Essays Based on Centering Theory. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*. Springer Berlin Heidelberg, Berlin, Heidelberg, pages 450–461. Cited on page 17.
- Vasile Rus and Nobal Niraula. 2012b. Automated detection of local coherence in short argumentative essays based on centering theory. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, pages 450–461. https://link.springer.com/chapter/10.1007/978-3-642-28604-9_37. Cited on pages 19, 31, and 120.
- D.Y. Sakhare and Raj Kumar. 2014. Syntactic and Sentence Feature Based Hybrid Approach for Text Summarization. *International Journal of Information Technology and Computer Science* 6:38–46. <https://pdfs.semanticscholar.org/3596/38997f8e9f8e6ad4edef6706ede8bceb9471.pdf>. Cited on page 62.
- Ted JM Sanders, Wilbert PM Spooren, and Leo GM Noordman. 1992. Toward a taxonomy of coherence relations. *Discourse processes* 15(1):1–35. Cited on pages 16 and 25.
- Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2019. A Hierarchical Multi-Task Approach for Learning Embeddings from Semantic Tasks. In *The Thirty-Third*

- AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019.* pages 6949–6956. <https://doi.org/10.1609/aaai.v33i01.33016949>. Cited on page 46.
- Deborah Schiffrin. 1987. *Discourse markers*. 5. Cambridge University Press. Cited on page 25.
- Rico Sennrich and Barry Haddow. 2016. Linguistic Input Features Improve Neural Machine Translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*. Association for Computational Linguistics, Berlin, Germany, pages 83–91. <https://www.aclweb.org/anthology/W16-2209>. Cited on page 62.
- Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, pages 2931–2951. <https://www.aclweb.org/anthology/P19-1282>. Cited on page 48.
- M Shermis and B Hammer. 2012. Contrasting state-of-the-art automated scoring of essays: analysis. In *Annual National Council on Measurement in Education Meeting*. pages 1–54. Cited on page 120.
- Haoyue Shi, Jiayuan Mao, Tete Xiao, Yuning Jiang, and Jian Sun. 2018. Learning Visually-Grounded Semantics from Contrastive Adversarial Samples. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, pages 3715–3727. <https://www.aclweb.org/anthology/C18-1315>. Cited on page 48.
- Xing Shi, Inkit Padhi, and Kevin Knight. 2016. Does String-Based Neural MT Learn Source Syntax? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1526–1534. <https://doi.org/10.18653/v1/D16-1159>. Cited on page 108.
- H. Gregory Silber and Kathleen F. McCoy. 2002. Efficiently Computed Lexical Chains as an Intermediate Representation for Automatic Text Summarization. *Computational Linguistics* 28(4):487–496. <https://doi.org/10.1162/089120102762671954>. Cited on pages 16 and 29.
- N Clayton Silver and William P Dunlap. 1987. Averaging correlation coefficients: Should Fisher’s z transformation be used? *Journal of Applied Psychology* 72(1):146. Cited on page 51.

- Karin Sim Smith, Wilker Aziz, and Lucia Specia. 2016a. Cohere: A Toolkit for Local Coherence. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), Portorož, Slovenia, pages 4111–4114. <https://www.aclweb.org/anthology/L16-1649>. Cited on page 49.
- Karin Sim Smith, Wilker Aziz, and Lucia Specia. 2016b. The Trouble with Machine Translation Coherence. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*. pages 178–189. <https://www.aclweb.org/anthology/W16-3407>. Cited on page 20.
- Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. pages 129–136. Cited on page 40.
- Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 231–235. <https://www.aclweb.org/anthology/P16-2038>. Cited on pages 46 and 60.
- Swapna Somasundaran, Jill Burstein, and Martin Chodorow. 2014. Lexical Chaining for Measuring Discourse Coherence Quality in Test-taker Essays. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, Dublin, Ireland, pages 950–961. <https://www.aclweb.org/anthology/C14-1090>. Cited on pages 16, 18, 19, 29, 99, and 120.
- Radu Soricut and Daniel Marcu. 2006. Discourse Generation Using Utility-Trained Coherence Models. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*. Association for Computational Linguistics, Sydney, Australia, pages 803–810. <https://www.aclweb.org/anthology/P06-2103>. Cited on pages 30, 32, and 99.
- Nicola Stokes, Joe Carthy, and Alan F Smeaton. 2004. SeLeCT: a lexical cohesion based news story segmentation system. *AI communications* 17(1):3–12. Cited on page 29.
- Michael Strube and Udo Hahn. 1999. Functional Centering: Grounding Referential Coherence in Information Structure. *Comput. Linguist.* 25(3):309–344. <http://dl.acm.org/citation.cfm?id=973321.973328>. Cited on page 27.

- Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. JMLR.org, ICML'17, pages 3319–3328. <http://dl.acm.org/citation.cfm?id=3305890.3306024>. Cited on page 47.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. MIT Press, Cambridge, MA, USA, NIPS'14, pages 3104–3112. <http://dl.acm.org/citation.cfm?id=2969033.2969173>. Cited on pages 39, 43, and 75.
- Kaveh Taghipour and Hwee Tou Ng. 2016. A Neural Approach to Automated Essay Scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1882–1891. <https://www.aclweb.org/anthology/D16-1193>. Cited on pages 40, 50, 75, 120, 121, 123, 124, 125, 126, 131, and 132.
- Doina Tatar, Diana Inkpen, and Gabriela Czibula. 2013. Text segmentation using Roget-based weighted lexical chains. *Computing and Informatics* 32(2):393–410. Cited on page 29.
- Yi Tay, Minh C Phan, Luu Anh Tuan, and Siu Cheung Hui. 2018. SkipFlow: Incorporating neural coherence features for end-to-end automatic text scoring. In *Thirty-Second AAAI Conference on Artificial Intelligence*. <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/viewPaper/16431>. Cited on pages 19 and 121.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019a. BERT Rediscovered the Classical NLP Pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, pages 4593–4601. <https://www.aclweb.org/anthology/P19-1452>. Cited on pages 39 and 56.
- Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019b. What do you learn from context? Probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=SJzSgnRcKX>. Cited on page 56.
- Lucien Tesnière. 1959. *Éléments de Syntaxe Structurale*. Klincksieck, Paris. Cited on page 61.
- Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5 - RMSProp. *Technical report*. Cited on page 74.

- Dat Tien Nguyen and Shafiq Joty. 2017. A Neural Local Coherence Model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 1320–1330. <http://aclweb.org/anthology/P17-1121>. Cited on pages 17, 19, 20, 42, 62, 68, 72, 99, and 109.
- Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich Part-of-speech Tagging with a Cyclic Dependency Network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL '03, pages 173–180. <https://doi.org/10.3115/1073445.1073478>. Cited on page 76.
- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9:2579–2605. Cited on page 47.
- Teun A Van Dijk. 1979. Pragmatic connectives. *Journal of pragmatics* 3(5):447–456. Cited on pages 16 and 25.
- Teun A Van Dijk. 1980. The semantics and pragmatics of functional coherence in discourse. *Speech act theory: Ten years later* pages 49–65. Cited on pages 16 and 25.
- Teun A. van Dijk and Walter Kintsch. 1983. *Strategies of discourse comprehension*. Academic Press. Cited on pages 25 and 27.
- Vladimir Vapnik. 1995. *The nature of statistical learning theory*. Springer, USA. Cited on page 31.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. pages 5998–6008. Cited on page 35.
- Suzan Verberne, LWJ Boves, NHJ Oostdijk, and PAJM Coppen. 2007. Evaluating discourse-based answer extraction for why-question answering. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 735–736. Cited on page 20.
- Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2016. Order Matters: Sequence to sequence for sets. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. <http://arxiv.org/abs/1511.06391>. Cited on page 44.

- Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer Networks. In *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., pages 2692–2700. <http://papers.nips.cc/paper/5866-pointer-networks.pdf>. Cited on page 44.
- Marilyn Walker, Sharon Cote, and Masayo Iida. 1994. Japanese Discourse and the Process of Centering. *Comput. Linguist.* 20(2):193–232. <http://dl.acm.org/citation.cfm?id=972525.972528>. Cited on page 27.
- Liang Wang, Sujian Li, Yajuan Lü, and Houfeng Wang. 2017a. Learning to rank semantic coherence for topic segmentation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pages 1340–1344. Cited on page 20.
- Liang Wang, Sujian Li, Yajuan Lv, and Houfeng Wang. 2017b. Learning to Rank Semantic Coherence for Topic Segmentation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 1340–1344. <https://www.aclweb.org/anthology/D17-1139>. Cited on page 144.
- Tianming Wang and Xiaojun Wan. 2019. Hierarchical Attention Networks for Sentence Ordering. In *Proceedings of the AAAI Conference on Artificial Intelligence*. volume 33, pages 7184–7191. Cited on pages 20, 44, and 48.
- Yucheng Wang, Zhongyu Wei, Yaqian Zhou, and Xuan-Jing Huang. 2018. Automatic essay scoring incorporating rating schema via reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. pages 791–797. <https://www.aclweb.org/anthology/D18-1090.pdf>. Cited on page 121.
- Bonnie Webber, Matthew Stone, Aravind Joshi, and Alistair Knott. 2003. Anaphora and discourse structure. *Computational linguistics* 29(4):545–587. Cited on page 26.
- Bonnie Lynn Webber. 1988. Tense as discourse anaphor. *Computational Linguistics* 14(2):61–73. Cited on page 16.
- Sholom M Weiss and Casimir A Kulikowski. 1991. *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems*. Morgan Kaufmann Publishers Inc. Cited on page 50.
- Henry George Widdowson. 1978. *Teaching language as communication*. Oxford University Press. Cited on pages 16, 24, and 25.
- Sarah Wiegrefe and Yuval Pinter. 2019. Attention is not not explanation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and*

- the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics, Hong Kong, China, pages 11–20. <https://www.aclweb.org/anthology/D19-1002>. Cited on page 48.
- Peter Wiemer-Hastings and Arthur C Graesser. 2000. Select-a-Kibitzer: A Computer Tool that Gives Meaningful Feedback on Student Compositions. *Interactive Learning Environments* 8(2):149–169. [https://doi.org/10.1076/1049-4820\(200008\)8:2;1-B;FT149](https://doi.org/10.1076/1049-4820(200008)8:2;1-B;FT149). Cited on page 29.
- Ethan Wilcox, Roger P. Levy, and Richard Futrell. 2019. What Syntactic Structures Block Dependencies in RNN Language Models? In *Proceedings of the 41st Annual Meeting of the Cognitive Science Society*. <https://arxiv.org/abs/1905.10431>. Cited on pages 102 and 110.
- DM Williamson, Xiaoming Xi, and FJ Breyer. 2012. A framework for evaluation and use of automated scoring. *Educational Measurement: Issues and Practice* 31(1):2–13. <https://onlinelibrary.wiley.com/doi/full/10.1111/j.1745-3992.2011.00223.x>. Cited on page 120.
- Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. 2019. Pay Less Attention with Lightweight and Dynamic Convolutions. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=SkVhlh09tX>. Cited on page 41.
- Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation <https://arxiv.org/abs/1609.08144>. Cited on page 38.
- Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. Text Readability Assessment for Second Language Learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, San Diego, CA, pages 12–22. <https://doi.org/10.18653/v1/W16-0502>. Cited on page 19.
- Peng Xu, Hamidreza Saghir, Jin Sung Kang, Teng Long, Avishek Joey Bose, Yanshuai Cao, and Jackie Chi Kit Cheung. 2019. A Cross-Domain Transferable Neural Coherence Model. In *The 57th Annual Meeting of the Association for Computational Linguistics (ACL 2019)*. ACL. Cited on pages 17, 41, 42, 43, 72, 73, 80, and 99.
- Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019a. End-to-End Open-Domain Question Answering with BERTserini. In *Proceed-*

ings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations). Association for Computational Linguistics, Minneapolis, Minnesota, pages 72–77. <https://www.aclweb.org/anthology/N19-4013>. Cited on page 37.

Yongxin Yang and Timothy M. Hospedales. 2017. Trace Norm Regularised Deep Multi-Task Learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. <https://openreview.net/forum?id=rknkNR7Ke>. Cited on page 46.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. XINet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*. pages 5754–5764. Cited on page 37.

Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. 2016a. Multi-task cross-lingual sequence tagging from scratch <https://arxiv.org/abs/1603.06270>. Cited on page 46.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016b. Hierarchical Attention Networks for Document Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 1480–1489. <https://doi.org/10.18653/v1/N16-1174>. Cited on pages 39, 40, 48, 53, and 54.

Helen Yannakoudakis and Ted Briscoe. 2012. Modeling coherence in ESOL learner texts. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, pages 33–43. <https://www.aclweb.org/anthology/W12-2004.pdf>. Cited on pages 16, 19, 29, 120, and 122.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A New Dataset and Method for Automatically Grading ESOL Texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, pages 180–189. <http://dl.acm.org/citation.cfm?id=2002472.2002496>. Cited on pages 50, 64, 120, 121, 122, and 124.

Helen Yannakoudakis and Ronan Cummins. 2015. Evaluating the performance of Automated Text Scoring systems. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, Denver, Colorado, pages 213–223. <https://www.aclweb.org/anthology/W15-0625>. Cited on page 50.

- Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, pages 947–953. Cited on page 77.
- Yongjing Yin, Fandong Meng, Jinsong Su, Yubin Ge, Linfeng Song, Jie Zhou, and Jiebo Luo. 2020. Enhancing Pointer Network for Sentence Ordering with Pairwise Ordering Predictions. In *AAAI Publications, Thirty-Four AAAI Conference on Artificial Intelligence*. AAAI Press. Cited on page 44.
- Yongjing Yin, Linfeng Song, Jinsong Su, Jiali Zeng, Chulun Zhou, and Jiebo Luo. 2019a. Graph-based Neural Sentence Ordering. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, pages 5387–5393. <https://doi.org/10.24963/ijcai.2019/748>. Cited on page 20.
- Yongjing Yin, Linfeng Song, Jinsong Su, Jiali Zeng, Chulun Zhou, and Jiebo Luo. 2019b. Graph-based Neural Sentence Ordering. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, pages 5387–5393. <https://doi.org/10.24963/ijcai.2019/748>. Cited on pages 44 and 45.
- Su-Youn Yoon, Aoife Cahill, Anastassia Loukina, Klaus Zechner, Brian Riordan, and Nitin Madnani. 2018. Atypical Inputs in educational applications. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*. pages 60–67. <https://www.aclweb.org/anthology/N18-3008.pdf>. Cited on page 122.
- Jianfei Yu and Jing Jiang. 2016. Learning Sentence Embeddings with Auxiliary Tasks for Cross-Domain Sentiment Classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 236–246. <https://www.aclweb.org/anthology/D16-1023>. Cited on page 46.
- Haoran Zhang and Diane Litman. 2018. Co-Attention Based Neural Network for Source-Dependent Essay Scoring. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, New Orleans, Louisiana, pages 399–409. <https://www.aclweb.org/anthology/W18-0549>. Cited on page 121.
- Mo Zhang, Jing Chen, and Chunyi Ruan. 2016. Evaluating the Advisory Flags and Machine Scoring Difficulty in the e-rater® Automated Scoring Engine. *ETS Research Report Series* 2016(2):1–14. Cited on page 122.

- Muyu Zhang, Vanessa Wei Feng, Bing Qin, Graeme Hirst, Ting Liu, and Jingwen Huang. 2015. Encoding World Knowledge in the Evaluation of Local Coherence. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, pages 1087–1096. <https://www.aclweb.org/anthology/N15-1115>. Cited on pages 20 and 32.
- Yue Zhang, Qi Liu, and Linfeng Song. 2018. Sentence-State LSTM for Text Representation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, pages 317–327. <https://www.aclweb.org/anthology/P18-1030>. Cited on page 45.
- Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tiejian Liu. 2020. Incorporating BERT into Neural Machine Translation. In *International Conference on Learning Representations*. <https://openreview.net/forum?id=Hyl7ygStwB>. Cited on page 37.
- Xunjie Zhu, Tingfeng Li, and Gerard de Melo. 2018. Exploring Semantic Properties of Sentence Embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Melbourne, Australia, pages 632–637. <https://www.aclweb.org/anthology/P18-2100>. Cited on page 112.
- Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*. pages 19–27. Cited on page 38.
- Barret Zoph and Kevin Knight. 2016. Multi-Source Neural Translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 30–34. <https://www.aclweb.org/anthology/N16-1004>. Cited on page 47.
- Will Y Zou, Richard Socher, Daniel M Cer, and Christopher D Manning. 2013. Bilingual Word Embeddings for Phrase-Based Machine Translation. In *EMNLP*. Association for Computational Linguistics, Seattle, Washington, USA, pages 1393–1398. <https://www.aclweb.org/anthology/D13-1141>. Cited on pages 17, 37, and 131.
- Kaja Zupanc and Zoran Bosnić. 2017. Automated essay evaluation with semantic analysis.

Knowledge-Based Systems 120:118 – 132. <http://www.sciencedirect.com/science/article/pii/S0950705117300072>. Cited on page 120.

EXAMPLES FROM YAHOO POSTS

score	Example
low	The Airplane, without it we wouldn't have military aircraft, Leonardo De Vinci came up with the original flying machine, which made that during the Renaissance period, not sure why he made it, but he did. The Wright Brothers, made a plane that worked and are the ones recognized for inventing it. The year for that is 1901. I would like to see some sort of machine that you can put on your head and whatever image you are seeing can be projected, I think it would be great for the law, because even if a person is lying they will still be thinking of it and the law would be able to see that.
medium	Having a free press is one of the greatest freedoms in the world. That said, we have to remember that the press is staffed by human beings, who aren't perfect. As such, the public should be more responsible in gathering facts from multiple sources. Fines could be levied for the most serious of breaches, but I think for the most part, media outlets do a good job of policing their own. To Dan Rather's credit, he did say that he did not purposely report bad facts, but failed to follow up on the story to see if it had any credibility and promptly resigned. I believe that news anchors are pushed to report news as fast as possible because in this world of 24/7 reporting, news has become even more ratings driven and the pressure to get one up on the competition is immense.
high	My husband and I lived together for a year and a half before we decided to get married, and we only decided to then because of the legalities involved after I became pregnant. If two people love one another, they don't need the government or a religious organization to sanction the relationship, and since 50% of American marriages end in divorce anyway, I think it's silly for people to hang onto a tradition that we've clearly out-lived. If getting married makes you happy, go for it. But I never felt like I needed a marriage certificate to validate my relationship with my husband—and it's been twenty years since we originally moved in together. What's absolutely terrifying to me is that many states are passing laws preventing couples from cohabitating without marriage. If we aren't careful, the christian right will have us all living in Victorian England again...perish the thought!

Table A.1: Examples from Yahoo Posts.

EXAMPLES FROM THE CLAD

Original	Sterling is now 0.7% lower against the dollar, after Downing Street told the media that a Brexit deal was “essentially impossible”. It’s trading at \$1.2207 against the dollar, and down 0.74% against the euro at €1.1122.
Swap	It’s trading at \$1.2207 against the dollar, and down 0.74% against the euro at €1.1122. Sterling is now 0.7% lower against the dollar, after Downing Street told the media that a Brexit deal was “essentially impossible”.
Prefix Insertion	<i>In specific</i> , it’s trading at \$1.2207 against the dollar, and down 0.74% against the euro at €1.1122. Sterling is now 0.7% lower against the dollar, after Downing Street told the media that a Brexit deal was “essentially impossible”.
Lexical Substitution	<i>The currency</i> is trading at \$1.2207 against the dollar, and down 0.74% against the euro at €1.1122. Sterling is now 0.7% lower against the dollar, after Downing Street told the media that a Brexit deal was “essentially impossible”.
Random	1- Sterling is now 0.7% lower against the dollar, after Downing Street told the media that a Brexit deal was “essentially impossible”. They can include a combination of up to seven different plastics as well as metal, and many hangers end up in landfill where they can take up to 1,000 years to break down, according to hanger recycling company First Mile. 2- Current plastic hangers are hard to recycle because of how they are made. It’s trading at \$1.2207 against the dollar, and down 0.74% against the euro at €1.1122.
Lexical Perturbations	Sterling is now 0.7% lower against the dollar, after Downing Street told the media that a Brexit deal was “essentially impossible”. It’s <i>looking</i> at \$1.2207 against the dollar, and down 0.74% against the euro at €1.1122.
Corrupt Pronoun	Sterling is now 0.7% lower against the dollar, after Downing Street told the media that a Brexit deal was “essentially impossible”. <i>He’s</i> trading at \$1.2207 against the dollar, and down 0.74% against the euro at €1.1122.
Original	Regional sales manager Kevin Navette is currently abroad, but he has been blocked by Vodafone from using his work phone. He told the BBC that he has been charged £3,000 and his service stopped working on Sunday.
Swap	He told the BBC that he has been charged £3,000 and his service stopped working on Sunday. Regional sales manager Kevin Navette is currently abroad, but he has been blocked by Vodafone from using his work phone.
Prefix Insertion	<i>On Tuesday</i> , he told the BBC that he has been charged £3,000 and his service stopped working on Sunday. Regional sales manager Kevin Navette is currently abroad, but he has been blocked by Vodafone from using his work phone.
Lexical Substitution	<i>The man</i> told the BBC that he has been charged £3,000 and his service stopped working on Sunday. Regional sales manager Kevin Navette is currently abroad, but he has been blocked by Vodafone from using his work phone.
Random	1- Regional sales manager Kevin Navette is currently abroad, but he has been blocked by Vodafone from using his work phone. It says it is “committed” to its relationship with the Post Office because customers will still be able to deposit cash. 2- Barclays has promised to freeze last-in-town and remote branch closures for two years. He told the BBC that he has been charged £3,000 and his service stopped working on Sunday.
Lexical Perturbations	Regional sales manager Kevin Navette is currently abroad, but he has been blocked by Vodafone from using his work phone. He told the BBC that he has been charged £3,000 and his <i>car</i> stopped working on Sunday.
Corrupt Pronoun	Regional sales manager Kevin Navette is currently abroad, but he has been blocked by Vodafone from using his work phone. <i>We</i> told the BBC that we have been charged £3,000 and our service stopped working on Sunday.
Original	American Airlines is extending cancellations of Boeing 737 Max flights until January as regulators continue to review proposed software changes to the grounded plane. It expects to gradually resume Max flights from 16 January 2020.
Swap	It expects to gradually resume Max flights from 16 January 2020. American Airlines is extending cancellations of Boeing 737 Max flights until January as regulators continue to review proposed software changes to the grounded plane.
Prefix Insertion	<i>However</i> , it expects to gradually resume Max flights from 16 January 2020. American Airlines is extending cancellations of Boeing 737 Max flights until January as regulators continue to review proposed software changes to the grounded plane.
Lexical Substitution	<i>The airline</i> expects to gradually resume Max flights from 16 January 2020. American Airlines is extending cancellations of Boeing 737 Max flights until January as regulators continue to review proposed software changes to the grounded plane.
Random	1- American Airlines is extending cancellations of Boeing 737 Max flights until January as regulators continue to review proposed software changes to the grounded plane. She is also the youngest recipient of the prize. 2- Prof Esther Duflo is only the second woman to win the Nobel prize in economics since it began in 1969. It expects to gradually resume Max flights from 16 January 2020.
Lexical Perturbations	American Airlines is extending cancellations of Boeing 737 Max flights until January as regulators continue to review proposed software changes to the grounded plane. It expects to gradually resume Max <i>lessons</i> from 16 January 2020.
Corrupt Pronoun	American Airlines is extending cancellations of Boeing 737 Max flights until January as regulators continue to review proposed software changes to the grounded plane. <i>He</i> expects to gradually resume Max flights from 16 January 2020.

Table B.1: Examples from the CLAD.

PEARSON’S AND SPEARMAN’S CORRELATIONS FOR THE ASAP DATASET

Model	Training Data	Pearson (r)								
		1	2	3	4	5	6	7	8	Avg
LSTM _{T&N}	Original	0.775	0.719	0.719	0.828	0.846	0.855	0.812	0.623	0.782
LSTM _{T&N-comb}	Combined	0.640	0.661	0.541	0.701	0.710	0.553	0.542	-0.446	0.524
LSTM _{T&N+ELMo}	Original	0.827	0.700	0.713	0.800	0.807	0.830	0.775	0.707	0.774
LSTM _{T&N+BERT}	Original	0.840	0.714	0.712	0.800	0.818	0.831	0.791	0.737	0.785
JL-LC	Combined	0.782	0.693	0.710	0.818	0.839	0.846	0.796	0.567	0.768
JL-LC _{no.layer_sharing}	Combined	0.783	0.658	0.712	0.823	0.843	0.840	0.789	0.581	0.766
JL-LC _{lstm_sharing}	Combined	0.805	0.666	0.694	0.818	0.841	0.837	0.755	0.558	0.760
JL-MTL _{GRs}	Combined	0.816	0.687	0.704	0.815	0.842	0.837	0.770	0.673	0.775
JL-MTL _{GRs+ELMo}	Combined	0.816	0.717	0.680	0.795	0.813	0.807	0.766	0.703	0.766
JL-MTL _{GRs+BERT}	Combined	0.825	0.711	0.674	0.798	0.800	0.824	0.781	0.738	0.773
Model	Training Data	Spearman (ρ)								
		1	2	3	4	5	6	7	8	Avg
LSTM _{T&N}	Original	0.732	0.687	0.719	0.835	0.848	0.839	0.804	0.605	0.770
LSTM _{T&N-comb}	Combined	0.720	0.678	0.619	0.716	0.753	0.678	0.622	-0.455	0.588
LSTM _{T&N+ELMo}	Original	0.821	0.706	0.713	0.801	0.813	0.814	0.792	0.703	0.774
LSTM _{T&N+BERT}	Original	0.835	0.716	0.712	0.804	0.819	0.809	0.801	0.734	0.782
JL-LC	Combined	0.753	0.663	0.710	0.824	0.843	0.826	0.785	0.539	0.756
JL-LC _{no.layer_sharing}	Combined	0.747	0.637	0.710	0.830	0.847	0.819	0.782	0.558	0.754
JL-LC _{lstm_sharing}	Combined	0.773	0.639	0.695	0.825	0.843	0.814	0.745	0.531	0.747
JL-MTL _{GRs}	Combined	0.812	0.701	0.706	0.823	0.846	0.826	0.786	0.667	0.778
JL-MTL _{GRs+ELMo}	Combined	0.808	0.727	0.681	0.799	0.816	0.784	0.782	0.691	0.765
JL-MTL _{GRs+BERT}	Combined	0.817	0.707	0.673	0.801	0.803	0.803	0.791	0.731	0.770

Table C.1: Results on the ASAP original test set; the first half reports Pearson’s (r) and the second reports Spearman’s (ρ) correlations. Each half is horizontally divided into two partitions to visually discriminate between AES models and joint models, in that order. For each model, I display the training dataset and Pearson or Spearman across the 8 prompts. In the final column, I report the average across all the prompts after applying Fisher transformation (§2.7).

THRESHOLDS FOR ADVERSARIAL DETECTION

Prompt	1	2	3	4	5	6	7	8
Score Range	2-12	1-6	0-3	0-3	0-4	0-4	0-30	0-60
JL-LC	7	3	2.5	2.5	3.5	3.5	15	30
JL-LC _{no_layer_sharing}	7	2.5	2.5	2.5	3.5	3.5	15	20
JL-LC _{lstm_sharing}	7	3	2.5	2.5	3	3.5	15	30
JL-MTL _{GRs}	7	3	2.5	2.5	3.5	3.5	20	40
JL-MTL _{GRs} +ELMo	7	2	2.5	2.5	3.5	3.5	15	30
JL-MTL _{GRs} +BERT	7	2.5	2.5	2	3.5	3.5	19	17

Table D.1: Thresholds for adversarial detection fine-tuned for each model on each prompt. The model flags an essay as adversarial if the difference between the predicted essay score and coherence score is greater than or equals this threshold.