Identifying Problem Statements in Scientific Text

Kevin HEFFERNAN, Simone TEUFEL
University of Cambridge
Computer Laboratory, 15 JJ Thomson Avenue, Cambridge CB3 0FD
forename.surname@cl.cam.ac.uk

Abstract. In this work, we focus on the automatic identification of fine-grained problem-solution structure in scientific argumentation. We operationalise the task of finding problem formulations within scientific text in a supervised setting, using a newly-created hand-curated corpus from the domain of computational linguistics. In terms of linguistic features for their detection, we distinguish features from within the statement, and features representing the surrounding context. Results from a classification task on our corpus show that the task of identifying problem statements is tractable using a mixture of features, whereby features modelling the rhetorical context are particularly successful. Overall, our experiment shows promise for future work in identifying scientific problem-solution structure in a more global way.

Keywords. problem-solving, argumentation, argumentative zoning

1. Introduction

Argumentation is a human activity that can take many shapes and forms. Almost every aspect of our life is governed by communicative needs to persuade somebody of something. The cognitive tasks associated with this have therefore left their traces in almost any extant written or transcribed textual material we as computational linguists might choose for automatic analysis and interpretation.

However, academic study of argumentation has been heavily biased towards areas of human argumentation that are associated with professional activity rather than private ones, and in particular those areas that are seen to be more "objective", such as political speech, legal contracts, and scientific articles. Following Aristotle, there has been a tradition to consider mainly arguments that are logically truth-conditional. More recently, computational linguists' attention has turned to defeasible arguments, i.e. those that people actually use in everyday argumentation, whether they are logically sound or not. Seen this way, interpreting arguments has more to do with assessing plausibility than with formal proof.

In this paper, we will present evidence for one particular facet of argumentation in science – problem-solution structure. The view of science as a problem-solving activity is a common assumption amongst many researchers [1,2,3,4,5]. [6] identified three basic types of scientific article: the "controlled experiment", the "hypothesis testing" and the

"technique description". Each type has its own structure, but according to [7] they can be reduced, either by degradation or by amelioration, to a problem-solution structure.

In earlier work, one of us presented a theory of argumentation moves in science (Argumentative Zoning; [8]), which can be operationalised as a supervised machine learning task that assigns a small number of rhetorical labels to individual sentences. Problem-solution structure is "hard-wired" into the labels and features a way of defining the task. For instance, the fact that an author declares a particular state of the world as "problematic" in a sentence might well lead to that sentence being classified as a research gap (the declared motivation for the knowledge claim that constitutes the paper). At the same time, linguistic features that might indicate problem-status (such as lexical items, the use of the verb "need", negative-polarity adjectives or negated verbs expressing solutionhood) will be detected in such a sentence if they are expressed in an explicit enough manner; this will eventually serve to classify the sentence as "CTR" (the label associated with a research gap).

However, Argumentative Zoning (AZ) treats several aspects of scientific argumentation simultaneously and collates all these phenomena into only 7 (or 12, in follow-on work [9]) classification labels. In contrast, we are interested in detecting descriptions of problems as a separate task here. The task we are setting ourselves in the current paper also differs by its formal definition. Rather than classifying an entire sentence, we will classify shorter linguistic strings extracted from the sentence, which might or might not describe a problem.

In this paper, we present (in section 2.1) the development of a small hand-curated training and testing corpus for a binary problem classification of real-world strings from articles in computational linguistics. We use explicit cue phrases to create this corpus, but hope to be able to apply the classifier to *any* problem description in scientific articles, whether explicit or not. Section 4 will present the results of a supervised machine learning experiment to replicate this classification. We split our features into those internal to the string and those using context around the candidate string. One of the core claims in AZ was that rhetorical labels of certain statements influence each others' rhetorical status; our experiments allows us to quantify this effect, as opposed to the effect of the semantics of the potential problem description itself.

1.1. Linguistic Correlates of Problem-hood

Let us now look at what a description of a problem might look like. A priori, we would expect any description of the body of scientific knowledge or the state of the world in general which is seen as negative. We count as problems descriptions of impracticality, lack of knowledge or of a failure of an existing attempt to rectify such a situation, i.e., an unsuccessful attempt of solving a problem. In this category, we also include statements where a solution unearthed follow-on problems. We also include all task descriptions as problems, i.e., all statements of tasks the authors are setting out to do in the current paper. The phrases we consider can syntactically be noun phrases, verb phrases, propositional phrases, and any other syntactic constituents determined by our parser, as long as they pass a human quality test (cf. section 2.1).

We will now discuss possible linguistic correlates of problem-solutionhood structure. Since descriptions of problems have a strong correlation with negative sentiment, identifying the polarity status of the head of each candidate phrase should intuitively help

in resolving a candidate's problem-hood. For example, in the phrase: "a complication", the head noun here ("complication") clearly identifies this statement as problematic. The syntactic characteristics of a candidate phrase should also help in classifying their status. Since problems are often posed as questions, this observation might be captured with WH- POS tags. Additionally, descriptions of problems often have a large proportion of adjectives or adverbs to qualify their badness (e.g. "the negatively skewed distribution"). Tense, negation and modality also play a role in determining sentiment. Making use of tense is an important aspect to consider when modelling an author's viewpoint. For example, previous work will be cited because it motivated something in the paper. However, it may be cited for use as a method (praise) or as a motivation (dismissal). Negation has been a popular technique shown to improve sentiment classification [10] where the intuition is that any word following a negation (e.g. "not") should be given a negative weight. Modality can also identify the mood of a statement [11] or hedging [12] and so we also took this into consideration. Since many words in our statements may not have a known polarity status, instead of using a null value, the semantic similarity of nouns or verbs in the candidate phrase to those with a known polarity status should help increase our success. Lastly, knowledge of the rhetorical context surrounding a candidate phrase should aid in determining its problem-hood.

We will model each of these linguistic features in section 3. We will now explain our experimental setup (how the corpus was created, and how the experiment was designed).

2. Method

2.1. Corpus

Our new corpus is a subset of the latest version of the ACL anthology released in March, 2016[1] which contains 22,878 articles in the form of PDFs and OCRed text. The 2016 version was also parsed using ParsCit [13]. ParsCit recognises not only document structure, but also bibliography lists as well as references within running text. A random subset of 2,500 papers was collected covering the entire ACL timeline. In order to disregard non-article publications such as introductions to conference proceedings or letters to the editor, only documents containing abstracts were considered. We preprocessed the corpus using tokenisation, sentence splitting and syntactic parsing with the Stanford Parser [14].

In order to define an indisputable ground truth for problem strings, we use textual templates such as "problem is X". These were executed using tregex and tsurgeon [15], a set of tools for structural search in trees and tree manipulation. An example of one of our templates is shown in Figure 1. To increase our recall of different-worded problem descriptions, we additionally use target words which are semantically close to the noun "problem". Semantic similarity was defined by training a deep learning distributional model using Word2Vec [16] on 18,753,472 sentences from a biomedical corpus based on all full-text Pubmed articles [17]. From the 200 semantically closest words to "problem" (decided by cosine similarity with our Word2Vec model), we manually selected 28 clear and unambiguous synonyms for use in the templates. Of the sentences matching the templates, 600 were randomly selected, and the syntactic phrase corresponding to X was

[1] http://acl-arc.comp.nus.edu.sg/

excised from the sentence. Both the template match and the problem phrase X itself were then plausibility-checked by two annotators without communication between them (the two authors of this paper).

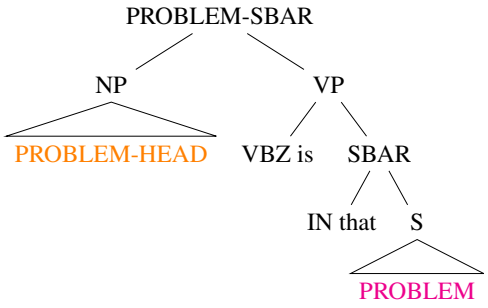

Figure 1. Template for PROBLEM-SBAR. PROBLEM-HEAD indicates the head noun of the NP must be one of our chosen problem words. Example: "The problem is that we do not achieve a significant result."

We also wanted to find similarly shaped negative examples, i.e., guaranteed non-problem strings. We sampled a population of phrases to mimic our 600 problem strings as closely as possible while making sure they really are negative examples. We started from sentences *not* containing any problem words (i.e. those used in problem templates). From each, we at random selected one syntactic subtree; from those we selected 600 that satisfy the following conditions: first, the distribution of the head POS tags of the non-problem strings perfectly matches the head POS tags[2] of the problem strings perfectly. Secondly, the distribution of the lengths of the non-problem strings must not be significantly different to that of the problem strings, using the Pearson's chi-squared goodness of fit test [19].

A human quality-test was then performed on problem and non-problem statements separately. Given a candidate problem statement within a sentence, the candidate was marked as positive if the string represented one of the following:

1. an unexplained phenomenon or a problematic state in science; or
2. a research question or a description of a task; or
3. an artifact that does not fulfil its stated specification.

Additionally, the lexical material inside the candidate string must not explicitly mark its status as a problem (e.g. "problem" or "difficult" must not appear *inside* in the string). We made this decision as such explicit signals would detract from the real task, that of judging the semantics of the string itself as problematic, without requiring explicit signals.

For each candidate non-problem statement, the candidate was marked positive if it conformed to both of the following rules:

1. The string is neither a phenomenon, a problematic state, a research question or a nonfunctioning artefact.
2. If the string expressed a research task, without explicit statement that there was anything problematic about it, we allowed for it to be defined as a non-problem.

[2]The head POS tags were found using the Collins' head finder [18].

Additionally, there must not be a different other description of a problem in the rest of the sentence (i.e. in the lexical items around the candidate). Non-grammatical/syntactic sentences were excluded (these could appear in our corpus as a result of its source being OCRed text). If the annotator found that the sentence had been slightly misparsed, but did contain a non-problem or problem, they were allowed to move the boundaries for the candidate string. This resulted in cleaner text, e.g., in the frequent case of coordination, when non-relevant constituents could be removed. This quality-test was conducted by both authors independently. From the set of sentences where both annotators agreed, 500 problem and 500 non-problem statements were randomly chosen.

The scientific documents containing statements resulting from the quality-test were converted to SciXML [8]. An AZ [20] model was trained on 80 computational linguistics papers (mutually exclusive to our quality-test document collection) which then predicted AZ [20] zones for each document.

2.2. Feature Extraction

To construct our feature sets, we began with a bag of words baseline using only the words within candidate phrases. This will tell us about the disambiguation ability of the problem description's semantics alone. Polarity of known words was then taken into account by first finding the head of each candidate phrase and then performing word sense disambiguation of each head using the Lesk algorithm [21]. We then looked up the polarity of the resulting synset using SentiWordNet [22]. Tense, negation, and modality were then added. To model negation, we specified a small set of negative words (e.g. "not") and for each word following a negation, appended "_not" until a phrase marking (e.g. ".,?"). Syntactic features were then added by including the POS tag distribution. We were careful not to base our model only on the head POS tag and the length of each candidate phrase, as these were features used for generating the non-problem candidate set. Since some phrasal heads may have been left without a sense by the Lesk algorithm (and thus with value NONE), we decided to use distributional semantic similarity between all nouns and verbs in each candidate phrase to words with a known polarity. We chose the words "poor" and "excellent" as these have been shown to be good indicators of polarity status in previous studies [23,24]. Semantic distance was calculated as before (cf. Section 2.1) using cosine similarity with our Word2Vec model. To take the rhetoric context into account, we used the AZ zones of the four sentences prior to each candidate phrase. However, when calculating the context of candidate sentences in the main body we never included the abstract, as these two sections fulfill different rhetorical functions and should not affect each other. For all features we decided not to use the additional textual material (other than the candidate phrase) contained in the sentence itself. This is done in order not to distort the task's difficulty.

3. Results

As shown in Figure 2, the bag of words baseline we chose performs better than random. Adding in the polarity of known synsets provides a small improvement, as does tense, negation, and modality. However, making use of the syntax within each candidate phrase provides a significant increase in performance. This may be due our obser-

Feature Sets	Classification Accuracy	
	NB	LR
$\text{Baseline}_{\text{bow}}$	57.1	56.7
+Polarity	55.6	56.9
+Tense, Neg, Mod	57.1	58.8
+Syntax	61.7*	65.6*
+Word2Vec	81.0*	84.5*
+AZ	81.4	84.7

Figure 2. Performance statistics for our classification task using Naïve Bayes (NB) and Logistic Regression (LR). Each consecutive feature set is cumulative. 10-fold cross-validation was used across all experiments. * denotes significance with respect to the previous feature set.

IG	Feature
0.7199	Word2Vec:poor
0.2437	Word2Vec:excellent
0.0258	pos:VB
0.0184	pos:.
0.0147	pos:JJ
0.0119	pos:DT
0.0112	pos:IN
0.0109	pos:TO
0.0104	pos:NNS
0.0103	pos:PRP
0.0102	pos:CD
0.0089	pos:WDT

Figure 3. Information gain (IG) in bits of top features from the best performing model (AZ).

vation that problems often take the form of questions, giving rise to a high concentration of WH- POS tags. Another significant performance increase was caused by using the Word2Vec model. This improvement is likely due to the effect of smoothing mentioned earlier: instead of receiving a null score for unknown words using the synset polarities in SentiWordNet, we are given a distance measurement. The marked improvements from Word2Vec are reflected in Figure 3, where Word2Vec attributes have the greatest information gain.

However, providing knowledge of the rhetoric context using the AZ zones leading up to each candidate statement provides the best performance for both classifiers used. This result supports one of the core claims of AZ: that rhetorical labels of certain statements influence each others' rhetorical status. Therefore, knowledge of the rhetorical context of a problem or non-problem is an important attribute for automatically classifying problem-solving structure within scientific argumentation.

4. Conclusions and Future Work

In this work, we have introduced a new hand-curated corpus of problem and non-problem statements, and shown that identifying and automatically classifying these statements is a tractable task. Our best system beat the baseline by a large margin, with the best performing feature set taking advantage of the statement's rhetorical context using Argumentative Zoning.

In future work, we intend to split the candidate statements into *tasks*, *problems* and *non-problems*. The fact that descriptions of tasks could be both a problem and non-problem in the rubric for hand-crafting our data set, is likely to provide a large degree of noise. Therefore, making this distinction may show a substantial increase in performance. We also plan to explore additional contextual features such as citations, and test the domain specificity of identifying problems and non-problems against corpora from other fields such as chemistry and genetics.

5. Acknowledgements

This work has been supported by the EPSRC. We thank the reviewers for their helpful comments.

References

[1] Michael Hoey. Signalling in discourse. 1979.

[2] Michael P Jordan. The rhetoric of everyday english texts. 1984.

[3] James P Zappen. A rhetoric for research in sciences and technologies. *New essays in technical and scientific communication*, pages 123–138, 1983.

[4] Bogdan Trawiński. A methodology for writing problem structured abstracts. *Information processing & management*, 25(6):693–702, 1989.

[5] VI Solovev. Functional characteristics of the authors abstract of a dissertation and the specifics of writing it. *Scientific and Technical Information Processing*, 3:80–88, 1981.

[6] Myrna Gopnik. *Linguistic structures in scientific texts*, volume 129. Mouton, 1972.

[7] John Hutchins. On the structure of scientific texts. *UEA Papers in Linguistics*, 5(3):18–39, 1977.

[8] Simone Teufel. The structure of scientific articles: Applications to citation indexing and summarization (center for the study of language and information-lecture notes). 2010.

[9] Simone Teufel, Advaith Siddharthan, and Colin Batchelor. Towards discipline-independent argumentative zoning: evidence from chemistry and computational linguistics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1493–1502. Association for Computational Linguistics, 2009.

[10] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326, 2010.

[11] Yang Liu, Xiaohui Yu, Zhongshuai Chen, and Bing Liu. Sentiment analysis of sentences with modalities. In *Proceedings of the 2013 international workshop on Mining unstructured big data using natural language processing*, pages 39–44. ACM, 2013.

[12] Ben Medlock and Ted Briscoe. Weakly supervised learning for hedge classification in scientific literature. In *ACL*, volume 2007, pages 992–999. Citeseer, 2007.

[13] Isaac G Councill, C Lee Giles, and Min-Yen Kan. Parscit: an open-source crf reference string parsing package. In *LREC*, 2008.

[14] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006.

[15] Roger Levy and Galen Andrew. Tregex and tsurgeon: tools for querying and manipulating tree data structures. In *Proceedings of the fifth international conference on Language Resources and Evaluation*, pages 2231–2234. Citeseer, 2006.

[16] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[17] Kathy McKeown, Hal Daume, Snigdha Chaturvedi, John Paparrizos, Kapil Thadani, Pablo Barrio, Or Biran, Suvarna Bothe, Michael Collins, Kenneth R Fleischmann, et al. Predicting the impact of scientific concepts using full-text features. *Journal of the Association for Information Science and Technology*, 2016.

[18] Michael Collins. Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4):589–637, 2003.

[19] Alan Agresti and Maria Kateri. *Categorical data analysis*. Springer, 2011.

[20] Simone Teufel et al. *Argumentative zoning: Information extraction from scientific text*. PhD thesis, Citeseer, 2000.

[21] Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, pages 24–26. ACM, 1986.

[22] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204, 2010.

[23] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

[24] Tony Mullen and Nigel Collier. Sentiment analysis using support vector machines with diverse information sources. In *EMNLP*, volume 4, pages 412–418, 2004.