Available online at www.sciencedirect.com
ScienceDirect
Computer Speech & Language 00 (2017) 1–15
www.elsevier.com/locate/csl

Source sentence simplification for statistical machine translation

Eva Hasler[a,*], Adrià de Gispert[a,b], Felix Stahlberg[a], Aurelien Waite[b], Bill Byrne[a,b]

[a] *University of Cambridge, Department of Engineering, CB2 1PZ Cambridge, U.K.*
[b] *SDL Research, Cambridge, U.K.*

Received 1 June 2016; received in revised form 17 October 2016; accepted 2 December 2016

Abstract

Long sentences with complex syntax and long-distance dependencies pose difficulties for machine translation systems. Short sentences, on the other hand, are usually easier to translate. We study the potential of addressing this mismatch using text simplification: given a simplified version of the full input sentence, can we use it in addition to the full input to improve translation? We show that the spaces of original and simplified translations can be effectively combined using translation lattices and compare two decoding approaches to process both inputs at different levels of integration. We demonstrate on source-annotated portions of WMT test sets and on top of strong baseline systems combining hierarchical and neural translation for two language pairs that source simplification can help to improve translation quality.

Keywords: Hierarchical machine translation; Text simplification; Neural machine translation

1. Introduction

Long sentences with complex syntactic structure and long-distance dependencies can be difficult to handle for translation systems. Even though systems that use syntax trees or other syntactic constraints on the source or target side can in theory alleviate these problems, in practise they often fail to outperform simpler models (Bojar et al., 2015).[1] Intuitively, shorter sentences are easier to translate as has been pointed out in the context of both traditional (Mellebeek et al., 2006; Xiong et al., 2009; Sudoh et al., 2010) and neural (Pouget-Abadie et al. (2014); Bahdanau et al., 2015) translation systems and we found empirical confirmation for this on our test sets where performance started degrading for sentences longer than 20 tokens. Even though the introduction of the attention mechanism for neural machine translation by Bahdanau et al. (2015) mitigates the effects of long input sentences, we believe there is still room for improvement in dealing with long and complex inputs.

Current translation systems have no notion of the relative importance of source tokens in long input sentences. However, many such sentences could be simplified by removing information that is not crucial to retain the central sentence meaning. For example, the additional information provided by relative clauses interrupts the fluency of the

* Corresponding author.
E-mail address: ech57@cam.ac.uk (E. Hasler).

[1] For example, phrase-based systems yield the best BLEU scores for German or Russian into English.

http://dx.doi.org/10.1016/j.csl.2016.12.001

main clause for the purpose of translation, while removing it would turn the higher-order structure as modelled in syntactic language models (Pauls and Klein, 2012) or dependency language models (Shen et al., 2008) into local phenomena. Additional, non-central information can also occur at the end of a sentence in the form of adverbials or coordinations which can make reordering decisions more difficult. For example, long range verb reordering in English to German translation may fail (Braune et al., 2012) or only be possible with low-scoring derivations.

Motivated by this, we investigate the following research question: if the machine translation system is provided with a simplified version of the input sentence (in addition to the original version), can it make good use of this information to improve translation quality? To study this question in detail, we make the following contributions. We manually create gold simplifications for three subsets of the English WMT test sets[2]. We take a *sentence compression* view to simplification, meaning that the simplified version is a strict substring of the input sentence. Section 3.1 describes the annotation procedure and provides examples. Our annotated data is released to the research community[3] to facilitate further research in translation out of English. We then propose two decoding strategies for a translation setup with original and simplified inputs (Section 4). The first approach is inspired by previous work on skeleton-based translation (Xiao et al., 2014) and performs a two-stage decoding process: first the simplified version is translated, and then the most likely hypotheses are used to constrain the decoding of the original version, thereby limiting the space of possible translation derivations. The second approach is less restrictive: both versions are decoded independently and the output lattices are combined so as to select only the full translation hypotheses that contain the simplified translation hypotheses as substrings. We evaluate the proposed strategies on top of strong English-to-German and English-to-French WMT baselines that combine hierarchical phrase-based and neural machine translation (NMT) very effectively. For English−German, we show modest gains in BLEU across 3 sets, but a closer analysis of the results reveals larger gains for those sentences where the simplified version has been obtained by extracting middle substrings of the original sentence. For English−French, we see larger overall gains which seem to be less dependent on the type of simplification. We conclude that both Hiero and neural models can benefit from explicit source simplification information.

A related research question is how to automatically obtain the simplified version of a sentence (McDonald, 2006) as input to a translation system. In this paper, we do not address this issue and leave it for future work; by releasing the annotated data we hope that new progress can be made along these lines.

2. Related work

Mellebeek et al. (2006) describe an early approach to skeleton-based translation, which decomposes input sentences into syntactically meaningful chunks. The central part of the sentence is identified and remains unaltered while other parts of the sentence are simplified. This process produces a set of partial, potentially overlapping translations which are recombined to form the final translation.

Sudoh et al. (2010) describe a “divide and translate” approach to dealing with complex input sentences. They parse the input sentences, replace subclauses with placeholders and later substitute them with separately translated clauses. Their method requires training translation models on clause-level aligned parallel data with placeholders in order for the translation model to deal with the placeholders correctly.

Xiao et al. (2014) build on the work by Mellebeek et al. (2006) but propose a simpler approach based on source skeletons used in a single decoding step. Their approach limits source sentence derivations to those consistent with the 1-best source skeleton, such that they can use an additional translation and language model score during decoding. They concluded that most of their gains came from a language model estimated on skeleton sentences with Xs marking the gaps. Pouget-Abadie et al. (2014) experiment with automatically segmenting the source sentence to overcome problems with overly long sentences. They show that segmenting the input is beneficial, but they do not consider the gappy input structures that can be created by source simplification.

Our work is related to multi-source translation where the sources are in different languages (Och and Ney, 2001; Zoph and Knight, 2016). While in our work, the second input is a variant of the first input, the idea that a second input provides some constraint to the translation search space is shared. Another variant of multi-source translation deals with multiple monolingual inputs which can be paraphrases of one another (Schroeder et al., 2009). This

[2] We use training and test data provided by the Workshop for Machine Translation (Bojar et al., 2015, WMT).
[3] http://dx.doi.org/10.17863/CAM.5868.

Please cite this article as: E. Hasler et al., Source sentence simplification for statistical machine translation, Computer Speech & Language (2017), http://dx.doi.org/10.1016/j.csl.2016.12.001

addresses issues with ambiguities or rare source words but does not deal with long-range dependencies, since all inputs are of comparable length.

A different approach to dealing with long-range dependencies are dependency language models (Shen et al., 2008; Sennrich, 2015) which can score non-adjacent parts of a translation hypothesis and mix terminal and non-terminal symbols. In contrast, skeleton-based translation deals with these dependencies in the input and does not rely on potentially noisy dependency structures built up during decoding.

Our work is most similar to the work by Xiao et al. (2014) but extends it in a few ways. First, we are comparing two different approaches to skeleton-aware translation to find out whether limiting the decoder to skeleton-consistent derivations is problematic. Second, we investigate the discriminative power of different models in place of Xiao et al.'s special language model and evaluate in the context of a stronger baseline system including neural models for machine translation. We also provide an analysis of the impact of different types of sentence skeletons on translation performance. Finally, Xiao et al. report on a translation task with Chinese as the source language to be simplified while our systems translate out of English.

3. Source sentence simplification

Sentence simplification is the process of altering a sentence such that it becomes simpler according to syntactic or lexical criteria. Existing approaches can be divided into those that perform simplification by deleting words and phrases (McDonald, 2006; Cohn and Lapata, 2007) and those that apply transformation rules to the input (Cohn and Lapata, 2008). Simplification by deletion or transformation is often referred to as *sentence compression*, where the percentage of retained tokens is given by the compression rate. Recent work has shown that this task can also be performed by an LSTM that predicts zeros or ones depending on whether an input token should be kept in the compression (Filippova et al., 2015). They showed that their model outperforms the compression model of McDonald (2006).

In this study, we are interested in simplification by deletion, and we call the result of applying it to a source sentence the *source sentence skeleton*. It is a simplified version of the translation input where only the most important parts of the sentence are retained and the remaining tokens are deleted. While there are automatic methods to produce simplified sentences, we focus on manually created source skeletons and leave the study of translation with automatic source skeletons for future work. We note however that there are realistic use scenarios in which fluent source language users of a translation system could highlight key portions of the text to help guide the SMT system.

3.1. Crowdsourcing sentence skeletons

We used the crowdsourcing platform Crowdflower[4] to produce gold skeleton annotations for three subsets of WMT test sets, each containing 1000 sentences between 20 and 40 tokens. To this end, we set up two tasks and collected three judgments each for all full sentences. We used the first task to gather skeleton data and passed it on to the second task to identify bad skeleton annotations from the first task. As simplification is an inherently ambiguous task, we do not expect or enforce annotators to agree on the same simplifications or compression rate, but rather aim to ensure grammaticality of the final simplifications.

Simplify by deletion. In this task, we asked workers to simplify a sentence by deleting words and punctuation, while trying to retain the most important information in the shortened sentence. We asked them not to change words, word order or add new words to the sentence and make sure the resulting sentence was still grammatical.

Identify bad simplifications. In this task, we presented workers with a full input sentence and three simplified versions of it. We asked them to mark all shortened sentences which were bad, because they were either ungrammatical or because they changed the basic structure of the full sentence or one if its clauses. We further instructed them that some information could be lost in the shortened sentences, which was intended, as long as the grammatical structure, especially the verbs, remained intact.

The output of the second task was aggregated using confidence scores of the workers to determine which of the three skeleton annotations per full sentence were of bad quality. We then chose one of the skeletons from the

[4] https://www.crowdflower.com.

Table 1
Compression rate and number of gaps/sentence for each of the annotated subsets.

Test subset	Compression rate	Avg gaps/sentence
newstest2012-subset	0.495	2.019
newstest2013-subset	0.481	1.623
newstest2014-subset	0.503	1.551

1. **the proposal** to remove article 365 from the code of criminal procedure , upon which the former prime minister was sentenced , **was supported by 147 members of parliament .**
2. in accordance with these plans , **the jazz and pop courses** , among others , **are to be relocated from the stuttgart music college to the mannheim music college .**
3. **this phenomenon gained momentum following the** november 2010 **elections** , which saw 675 new republican representatives added in 26 states **.**

Fig. 1. Examples of source sentences annotated with their simplified versions. The tokens in bold belong to the sentence skeleton and all remaining tokens fall into one or more gaps.

Table 2
Number of examples and compression rates for 3000 skeleton-annotated sentences with gaps in different locations. Middle: one or more gaps not at the start/end of sentence, Mixed: Middle + gap at start/end, Start: gap at the start, End: gap at the end, Brace: Start + End.

Gap type	No. examples (%)	Avg compr. rate
Middle	310 (10.3%)	0.625
Mixed	866 (28.9%)	0.467
Start	332 (11.1%)	0.581
End	824 (27.5%)	0.478
Brace	668 (22.2%)	0.444

remaining set. If the judgments from the second task were below our confidence threshold or no skeletons had been produced for an input sentence, we manually selected or produced a skeleton sentence to make sure each full sentence was annotated. The agreement on the second task averaged over all proposed skeletons (each a binary choice with three annotators) was $\sim 91\%$, thus in the majority of cases the workers agreed whether a proposed skeleton was of good quality. The compression rates and average gaps per sentence of the annotated subsets are given in Table 1.

In Fig. 1, we show three examples of a source sentence annotated with its simplified version (one of the possible sentence skeletons). For example, in the first sentence, the skeleton is what remains after removing the infinitive and relative clauses. This results in the noun phrase *the proposal* being adjacent to the verb complex *was supported* in the skeleton. The sentence can be viewed as having a long gap in the middle where part of the full sentence was removed, while the remaining beginning and end of the sentence have been left intact. However, we can also have sentences with multiple such gaps and they can be in different places as well, as shown in example 2 and 3 which have gaps in the beginning and end of the sentence. Note that alternative skeletons could be equally grammatical, for example the skeleton for the first sentence in Fig. 1 could be further simplified to *the proposal was supported*.

It is not entirely obvious what kinds of gaps would be most suitable for a skeleton to be used as input to a translation system, though it seems that gaps that make previously non-adjacent but dependent parts of the sentence adjacent would be most useful. Table 2 shows a breakdown of our 3000 annotated sentences by their gap types along with the average compression rates which will be used for further analysis in Section 6.1.1.

4. Machine translation with original and simplified inputs

We propose a general two-step decoding framework for translating with an additional source skeleton. In the first step, a decoder is used to translate the source skeleton and produce an *n*-best list of candidates. In the second step,

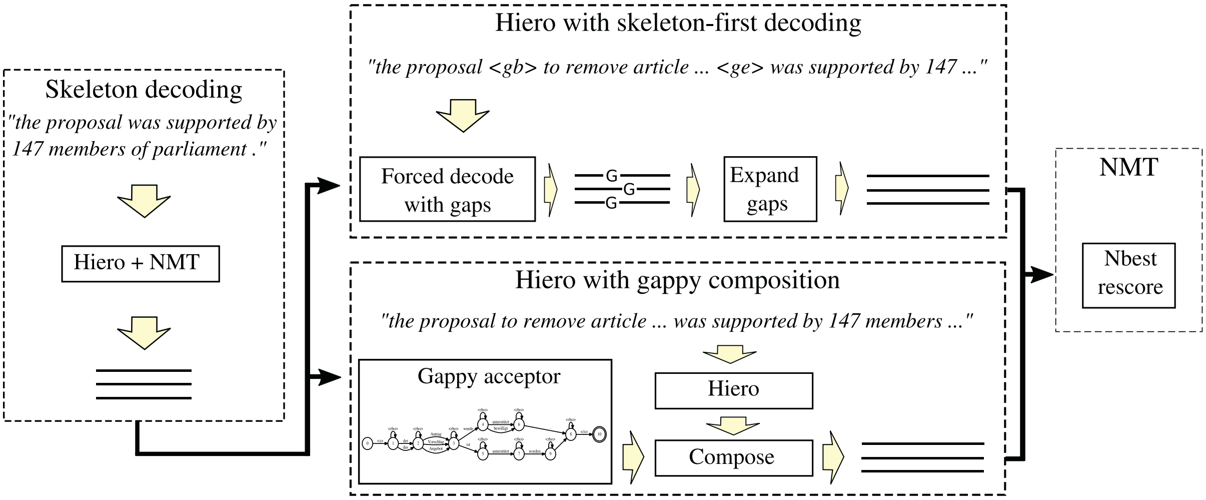
Hiero with skeleton-first decoding
"the proposal <gb> to remove article ... <ge> was supported by 147 ..."
Skeleton decoding
"the proposal was supported by 147 members of parliament ."
Hiero + NMT
Forced decode with gaps
G
G
G
Expand gaps
NMT
Nbest rescore
Hiero with gappy composition
"the proposal to remove article ... was supported by 147 members ..."
Gappy acceptor
Hiero
Compose

Fig. 2. Graphical representation of our decoding approaches.

this *n*-best is used to guide the decoding of the full sentence. The assumption is that getting the skeleton structure right in translation will have a positive impact on the full translation, because dependent tokens with string-level gaps can be closer together in the skeleton sentence (depending on the type of gap), and applying additional models to the skeleton provides new information which is not available when decoding the full source sentence. For gaps that do not introduce new *n*-grams, we expect benefits from the smaller search space, for example for reordering.

The second decoding step is challenging: it has to decide where the translations of the gaps are to be inserted in the skeleton output to produce the full translation hypothesis. We now compare two alternative approaches to perform this: *skeleton-first decoding* and *composition with gappy skeleton lattices*, both of which are depicted in Fig. 2. In this work, we use a Hiero system with neural MT rescoring as decoding framework, but other models could also be used. However, our proposed decoding framework does rely on lattice operations such as composition and partial expansion, so any alternative baseline system should ideally output translation lattices as well.

4.1. Skeleton-first decoding

When translating the full input sentence, one can restrict the Hiero decoder so that it outputs derivations that are consistent with the presence of gaps. This can be achieved by (a) marking the input sentences with skeleton information in the form of *gap start* and *gap end* symbols, and (b) making the Hiero decoder use a different non-terminal symbol (for example, G) than the usual X for rules that cover gap spans. This way, the decoder produces derivations which distinguish the skeletal and non-skeletal parts of the derivation because the non-skeletal parts will be headed by G symbols. In addition, we extend the grammar with rules that leverage contextual information from the skeleton. For example, if the grammar contains a phrasal rule that applies to the source skeleton but not to the full source sentence, we use the internal word alignments to insert G symbols at the respective positions in the source and target sides of the rule. If such rules exist for a given input sentence and skeleton, the decoder may be able to generate new hypotheses that would not have been part of the original Hiero lattice.

We then run the decoder so that it leaves all G-rooted spans unexpanded, thus producing skeleton hypotheses that contain G symbols in the output (see top of Fig. 2). These symbols are ignored when applying a language model or any other model (such as the neural model) to the partially expanded lattice. In practise, we implemented this decoding step as composition with skeleton lattices which can be seen as forced decoding towards the skeleton translation candidates while keeping gaps in the output strings. Finally, the gaps are subsequently expanded to produce the full hypotheses which are scored with a standard language model.

This process is inspired by Xiao et al. (2014), who use skeleton language and translation model scores in addition to the scores on full hypotheses. They restrict the derivations of the full source sentences to those consistent with the skeleton source sentences, i.e., derivations where the source sides of bilingual rules do not cross skeleton boundaries as denoted by gap start and end symbols at the top of Fig. 2.

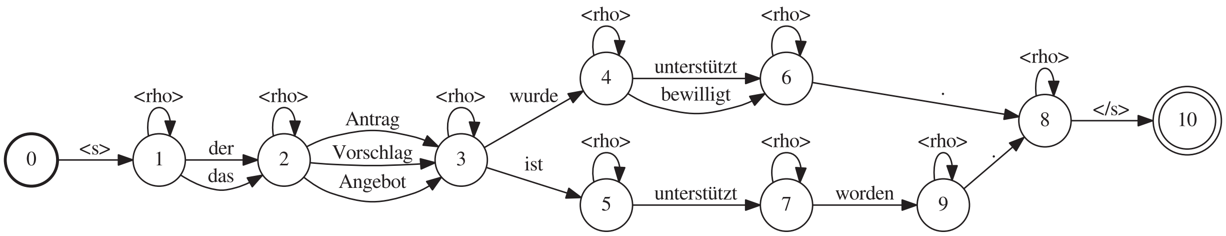

Fig. 3. Gappy acceptor for an *n*-best list of German skeleton translations for the English sentence *the proposal was supported*. Loops with <*rho*> symbols can consume any remaining symbols from the full Hiero lattice between tokens from the skeleton translations.

4.2. Composition with gappy skeleton lattices

Our second decoding approach is based on two separate full and skeleton decoding steps which are combined through lattice composition as shown at the bottom of Fig. 2. The motivation behind this approach is that we want to avoid limiting the full translation *derivations* to be consistent with the skeleton translation derivations. That is because depending on the positions of gaps in the source sentence, this requirement rules out phrasal translation rules spanning across gap boundaries. Gappy composition does not alter the hypothesis space for decoding with the full input, but instead retrieves those full hypotheses that are consistent with the skeleton *translations*. Therefore, contextual information in the full input is preserved. A potential limitation of this approach is that skeleton translations must be reachable by the decoder when operating on the full input. However, this could be addressed by allowing the Hiero decoder to generate more hypotheses with looser decoding parameters.

Gappy skeleton acceptors. In order for composition between full and skeleton lattices to be successful, we have to turn the skeleton lattices into gappy acceptors (Mohri, 2003), which accept each of the possible skeleton hypotheses in the skeleton lattice, interleaved with tokens from the non-skeletal spans of the full sentence. We achieve this by adding loops to all states in the skeleton lattice, with labels coming from the vocabulary of the full lattice or, equivalently, <rho> symbols which can consume all symbols that are not already on outgoing arcs from the same state (see Fig. 3). For efficiency, we apply some pruning on the full lattices before composition and prune the skeleton lattices to the *n*-shortest unique paths.

Model formulation. Let s be the full source sentence and s' its 1-best skeleton.[5] We are looking for the target sentence $t(\widehat{d})$, where $\widehat{d}$ is the highest scoring derivation of the full sentence, given both inputs s and s'. We denote as d' the derivation of a skeleton translation t', with $t' \equiv t(d')$. The probability of a target sentence is then modelled as the joint probability of a full target sentence t and its contained skeleton translation t', where we use the $\subset$ symbol to denote the substring relation, $t' \subset t$. This probability is expressed as a log-linear model using features associated with d, $\mathbf{f}(d)$, and features associated with d', $\mathbf{f}(d')$, and their respective feature weights $\mathbf{w}$ and $\mathbf{w}'$ as shown in Eq. (1). When adding skeleton features to the baseline features, the feature vectors provided by Hiero in the skeleton decoding step have the same dimensionality as the baseline feature vectors for the full input, while the NMT system provides a single score that expresses the skeleton translation quality. The translation $\widehat{t} \equiv t(\widehat{d})$ is found via

$$\begin{aligned}\widehat{d} &= \arg\max_{d} P(t|s,s') \\ &\approx \arg\max_{d} P(t|s)\cdot P(t'|s'), \quad s.t.\ t'\subset t \\ &\approx \arg\max_{d:\ t'\subset t(d)} (\mathbf{w}\cdot\mathbf{f}(d) + \mathbf{w}'\cdot\mathbf{f}(d'))\end{aligned} \tag{1}$$

This formulation differs from Xiao et al. (2014) in that the derivations for full and skeleton translations are decoupled from each other which allows for more modeling flexibility. While their system produces only those

[5] We could consider multiple skeletons per source sentence in which case we would aggregate the hypotheses resulting from the possible skeletons when computing translation probabilities. For the sake of simplicity and because we do not have multiple skeletons in all cases, we only used a single skeleton per sentence.

derivations which are consistent with the source sentence skeleton, we avoid this limitation through separate decoding steps. Disallowing rules that cross skeleton boundaries can potentially reduce the fluency around these boundaries even if the skeleton constitutes a well-formed sentence.

4.3. Fine-tuning with skeleton scores

Both approaches to decoding with additional skeleton information allow us to integrate the scores of external models such as the scores of separate skeleton lattices and neural language or translation models into our system. While we could directly interpolate the additional model score with the combined translation and language model score of the full translation hypotheses, we found it more beneficial to retune the weights of the baseline features together with the skeleton feature(s). Similarly, when NMT rescoring is applied, the features are retuned with the additional NMT feature included.

5. Experimental setup

Hiero baseline. Our first baseline is a hierarchical phrase-based translation system (Chiang, 2005; 2007; Iglesias et al., 2009) which outputs translation lattices. We trained an English–German system on the WMT 2015 training data (Bojar et al., 2015) comprising 4.2 M parallel sentences from the *Europarl, News Commentary v10* and *Commoncrawl corpora*. We word-aligned the parallel data using MTTK (Deng and Byrne, 2006) and extracted a *shallow-1* translation grammar with additional rules for verb movement (de Gispert et al., 2010). We use a 5-gram language model trained on portions of the parallel and monolingual target side data and interpolated to minimize perplexity on a development set (Schwenk and Koehn, 2008). The language models were trained and binarized using SRILM (Stolcke, 2002) and KenLM (Heafield et al., 2013). The system uses 12 standard translation and language model feature functions and the feature weights are tuned with lattice MERT (Macherey et al., 2008). The system performance is comparable to systems without neural language models that have been submitted to the Workshop for Machine Translation for the respective test sets. We trained an English–French system on WMT 2015 training data comprising 12.1 M parallel sentences in a similar fashion. We extended the translation grammar with provenance features (Chiang et al., 2011) because of the additional UN and Gigaword corpora for this language pair, resulting in 32 features in total.

NMT baseline. Our second baseline is the attention-based neural machine translation model presented by Bahdanau et al. (2015), using the Blocks implementation of van Merrienboer et al. (2015). We use the same English–German training data as for our Hiero baseline and train the model for $\sim$ 3 weeks on a Tesla K40 GPU (this includes the time for decoding the development set at certain intervals). We use a vocabulary of 50 k for the source and target side and do not deal with unknown output words. For English–French, we used the preprocessed training data of Schwenk (2014) which was created using data selection techniques and a vocabulary of 30 k. We augmented the data with truecasing to match our Hiero baseline. The training data and vocabulary for our neural baselines matches the baselines described in Jean et al. (2015). The standalone performance of these models is shown in Tables 3 and 5 when run with a beam of size 12.

Hiero + NMT baseline. Our third baseline system is a combination of Hiero and NMT similar to the system described in Stahlberg et al. (2016) who reported large gains from rescoring Hiero lattices with a neural translation system. In contrast to their work and because it is infeasible to score entire lattices, we instead score 1000-best lists with the NMT system. Thus, for all experiments involving NMT scores of full translation hypotheses, we score the 1000-best hypotheses under the respective model with the NMT system such that we can integrate and re-tune our system with the additional scores.

Skeleton translations. We use the Hiero baseline system to produce translation lattices for the skeleton input sentences. For lattice composition in the two decoding procedures described in Section 4, we either take a 1000-best list under the Hiero baseline model, along with the Hiero baseline features, or a 1000-best list under the NMT model with a single score. In the latter case, we use the method described by Stahlberg et al. (2016) to retrieve the *n*-best

Table 3
English–German results on annotated subsets of WMT dev/test sets. The reported gains refer to the system at the bottom of the table. Bolded numbers mark the best result in a given column while statistical significance is marked by * ($p \leq 0.05$).

System	**nt2012-subset**		**nt2013-subset**		**nt2014-subset**	
	Meteor	Bleu	Meteor	Bleu	Meteor	Bleu
Hiero	38.09	17.34	40.21	18.93	41.76	19.41
Nmt	34.11	15.31	36.97	17.85	36.55	16.61
Hiero (1000-best) + Nmt	39.53	19.12	41.78	20.77	43.09	20.87
Skeleton-first decoding						
Hiero + Hiero(skeleton)	38.10	17.57	40.14	18.83	41.80	19.36
Hiero + Nmt(skeleton)	38.60	18.00	40.74	19.71	42.44	20.37
Hiero + Nmt(skeleton) (1000-best) + Nmt	39.37	18.75	41.35	20.52	43.13	**21.06**
Gappy composition						
Hiero + Hiero(skeleton)	38.39	17.79	40.68	19.45	42.16	19.87
Hiero + Nmt(skeleton)	38.85	18.32	40.98	20.02	42.59	20.24
Hiero + Nmt(skeleton) (1000-best) + Nmt	**39.84**	**19.39**	**41.79**	**20.94**	**43.30**	20.99
Gain > Hiero (1000-best) + Nmt	+0.31*	+0.27	+0.01	+0.17	+0.21	+0.12

translation hypotheses in the skeleton lattice under the NMT model score. This method performs a beam search through the lattice while scoring translation prefixes with the NMT model.

Evaluation. We use *news-test2012* as our development set for Hiero and test on *news-test2013* and *news-test2014*. We report performance on subsets of 1000 sentences for which we have source skeletons from Crowdflower.[6] While these subsets consist of sentences with length between 20 and 40 tokens, the length restrictions were imposed solely to make annotation easier and do not imply algorithmic limitations of our approach, e.g., we would expect longer sentences to also benefit from simplification. For short sentences that do not require simplification, the respective decoding steps would simply be skipped.

We report Meteor scores (Banerjee and Lavie, 2005) and lowercased Bleu scores (Papineni et al., 2002) computed using the multi-bleu.pl script. Fine-tuning with the additional Hiero or NMT skeleton features and/or the additional NMT score for full hypotheses (1000-best rescoring) was carried out using lattice MERT (Macherey et al., 2008). For the English–German systems with added NMT scores (skeleton or full hypothesis), we found that the length ratios remained more stable when tuning on 1000 sentences selected from *news-test2012* and the respective other test set with an average source/reference length ratio close to 1. However, this had only a minor effect on the resulting scores and all remaining experiments were fine-tuned using the *news-test2012* subset.

6. Results

In this section, we present our translation results for English–German and English–French. We compare the performance of both decoding schemes described in Section 4 when adding skeleton features from the Hiero lattices or from the NMT system. We further apply 1000-best rescoring of full hypotheses with the NMT system to the Hiero baseline and to our proposed decoding schemes with skeleton information.

6.1. English–German results

The results of our English–German translation experiments are shown in Table 3. First of all, we can observe that in most cases, gappy composition with the skeleton output space yields better results than skeleton-first decoding which limits the translation derivations to those consistent with source skeleton boundaries. In comparison to the Hiero baseline, gappy composition yields small gains of up to $\sim$ 0.5 Meteor and Bleu with the combination of all Hiero features on full and skeleton hypotheses. This suggests that with good source skeletons, the skeleton output space does not restrict the full hypothesis space in any harmful way and that skeleton features can help to produce a better ranking of the full hypotheses. However, when we compose with an *n*-best list of skeleton translations scored

[6] Note that for English–French, we only have 555 skeletons for *news-test2014* because the test set overlaps only partially with the English–German test set for which we collected gold simplifications originally.

Table 4
English–German results according to Bleu for different systems, broken down by gap types as introduced in Section 3.1 (Sfd: skeleton-first decoding, Gc: Gappy composition).

System	Middle	Mixed	Start	End	Brace
Hiero	17.64	18.43	17.52	18.47	19.89
Sfd, Hiero + Nmt(skeleton)	19.24	19.11	18.23	19.39	20.29
Gc, Hiero + Nmt(skeleton)	19.46	19.62	18.38	19.34	20.16
Gain > Hiero	+1.82	+1.19	+0.86	+0.87	+0.27
Hiero (1000-best) + Nmt	19.78	20.31	18.78	20.05	21.48
Sfd, Hiero + Nmt(skeleton) (1000-best) + Nmt	20.15	20.17	18.92	20.04	21.00
Gc, Hiero + Nmt(skeleton) (1000-best) + Nmt	20.56	20.56	19.01	20.22	21.40
Gain > Hiero (1000-best) + Nmt	+0.78	+0.25	+0.23	+0.17	-0.08

by the NMT system, we can achieve substantially better results for both decoding schemes and across all test sets. This indicates that in order to make full use of the skeleton information, we need a stronger model to select fluent skeleton translations. While the NMT system on its own performs worse than Hiero, constraining it to the hypotheses in the skeleton lattice yields very useful skeleton *n*-best lists.

In a further experiment, we rescore the 1000-best Hiero lattices from normal decoding with the NMT model, shown as *Hiero (1000-best)* + Nmt in Table 3, and compare the result to rescoring the 1000-best full translation hypotheses resulting from our skeleton decoding schemes. While the skeleton-first decoding approach only outperforms this strong baseline on *news-test2014*, we see small but consistent gains for both Meteor and Bleu on all three sets for Hiero with gappy composition. It is worth noting that we achieve these translation results despite significantly reducing the space of hypotheses in the composed lattices. This shows that the search space constrained by the skeleton translations contains hypotheses which are at least as good as those found the in the original Hiero 1000-best lists when rescoring with the NMT system. This provides evidence that good source skeletons can be used to ensure consistency between full and skeleton translations without loss in overall quality.

6.1.1. Analysis of translation quality

In order to better understand the strengths and weaknesses of the skeleton-based translation approaches, we analyse our results by breaking down the annotated source sentences by the type of gap(s) in the skeleton, as introduced in Section 3.1. The translation quality on these subsets is shown in Table 4. While the results are slightly different depending on the decoding strategy (and mostly better for gappy composition), we observe that Bleu improvements over the Hiero baseline are largest when the skeleton has one or more gaps somewhere in the middle of the sentence (Middle, Mixed) and lowest for the Brace gap type, which removes context on the left and right. Note that the results in the top part of Table 4 constitute intermediate results after integrating the skeleton scores and before rescoring the full hypotheses with the NMT system. Therefore, these scores provide an insight into the effects of the different model components.

More importantly, we observe the same tendency regarding gap types in comparison to our stronger baseline with NMT scores on full hypotheses. On sentences with one or more gaps in the middle of the sentence, we see improvements of $\sim$ 0.8 Bleu with gappy composition. On other gap types, there are smaller improvements and a minor degradation for the Brace gap type. These results confirm our initial intuition that simplification is most helpful when it makes previously non-adjacent but potentially dependent tokens in the full sentence adjacent, as is the case for the Middle and Mixed gap types. The Brace gap type performs worst for both decoding schemes which could indicate that too much context is removed.

6.1.2. Translation examples

Fig. 4 shows an example of translation with the gappy composition approach. The syntactically correct (*n*-best) skeleton translation which reorders the translation of *plans to double* (*plant .. zu verdoppeln*) is contained in the full lattice and can be pulled out by gappy composition with the skeleton *n*-best list. For comparison, the Hiero baseline provides a bad translation of this verb complex (*Pläne für eine Verdopplung*). Fig. 5 shows an example of improved translation with gappy composition when we additionally score full hypotheses with the NMT. In this example, source simplification leads to a better translation and positioning of the source verb *search*, providing a more fluent translation than *Hiero (1000-best)* + Nmt which translates it into a noun.

Source, **skeleton source**	**the world market leader** in enterprise software ***plans to double*** **its number of offices** from five to ten or eleven **.**
Skeleton translation	der Weltmarktführer *plant* , die Zahl der Büros *zu verdoppeln* .
Hiero	der Weltmarktführer für Unternehmenssoftware *Pläne für eine Verdopplung* der Zahl der Büros von fünf auf zehn oder elf .
Hiero + Nmt(skeleton)	der Weltmarktführer für Unternehmenssoftware *plant* , die Zahl der Büros *zu verdoppeln* , von fünf bis zehn oder elf .

Fig. 4. Example of translation with the gappy composition decoding strategy. The English phrase *plans to double* is correctly translated to *plant .. zu verdoppeln* when composing the Hiero lattice with the skeleton *n*-best list which contains the above skeleton translation.

Source, **skeleton source**	**should someone *search* for the word** that would best express the imb strategy **, the closest one would** probably **be " long-term . "**
Skeleton translation	sollte jemand nach dem Wort *suchen* , das nächste wäre " langfristig " .
Hiero (1000-best) + Nmt	sollte jemand *Suche* nach dem Wort , das wäre am besten zum Ausdruck bringen das IMB - Strategie , die nächste wäre wahrscheinlich " langfristig " .
Hiero + Nmt(skeleton) (1000-best) + Nmt	sollte jemand nach dem Wort *suchen* , die am besten die IMB - Strategie zum Ausdruck bringen , das nächste wäre wahrscheinlich " langfristig " .

Fig. 5. Comparison of translations found by rescoring the 1000-best hypotheses from the original Hiero lattice with the NMT system or by rescoring the 1000-best of the composed lattice using both the full and skeleton NMT scores, using gappy composition.

6.2. Results for English−French

The results of our English−French translation experiments are shown in Table 5. For *news-test2013* and *news-test2014*, the performance of NMT according to Bleu is closer to Hiero than for English−German. However, similar to the English−German results the Meteor scores (which include scores for stems) are substantially lower for NMT, indicating many out-of-vocabulary words. Thus, even though the Bleu scores look comparable, the NMT system is not yet on par with Hiero. Like for English−German, gappy composition with NMT skeletons yields higher performance than skeleton-first decoding in the majority of cases and both decoding schemes yield significant improvements over the Hiero baseline when provided with an *n*-best list of skeleton translations selected according to the

Table 5
English−French results on annotated subsets of WMT dev/test sets. The reported gains are with respect to the system at the bottom of the table. Bolded numbers mark the best result in a given column while statistical significance is marked by * ($p \leq 0.01$).

System	**nt2012-subset**		**nt2013-subset**		**nt2014-subset**	
	Meteor	Bleu	Meteor	Bleu	Meteor	Bleu
Hiero	50.03	28.24	50.55	28.93	54.19	31.48
Nmt	45.64	25.84	47.15	28.13	51.10	31.93
Hiero (1000-best) + Nmt	51.66	30.76	52.49	31.70	56.83	35.38
Skeleton-first decoding						
Hiero + Nmt(skeleton)	50.55	29.00	51.42	30.41	55.88	33.79
Hiero + Nmt(skeleton) (1000-best) + Nmt	51.55	30.46	52.69	31.99	57.61	**36.36**
Gappy composition						
Hiero + Nmt(skeleton)	50.99	29.55	51.51	30.35	55.96	33.68
Hiero + Nmt(skeleton) (1000-best) + Nmt	**51.83**	**30.95**	**53.06**	**32.54**	**57.71**	36.28
Gain > Hiero (1000-best) + Nmt	+0.17	+0.19	+0.57*	+0.84*	+0.88*	+0.90*

Table 6
English–French results according to Bleu for different systems, broken down by gap types as introduced in Section 3.1 (Sfd: skeleton-first decoding, Gc: Gappy composition).

System	Middle	Mixed	Start	End	Brace
Hiero	27.56	30.41	28.82	27.65	30.61
Sfd, Hiero + Nmt(skeleton)	28.26	31.17	30.80	29.56	32.44
Gc, Hiero + Nmt(skeleton)	28.65	31.80	30.24	29.46	32.40
Gain > Hiero	+1.09	+1.39	+1.42	+1.81	+1.79
Hiero (1000-best) + Nmt	29.44	33.23	32.07	30.89	33.73
Sfd, Hiero + Nmt(skeleton) (1000-best) + Nmt	29.33	33.14	32.45	31.36	34.25
Gc, Hiero + Nmt(skeleton) (1000-best) + Nmt	30.28	33.82	32.46	31.63	34.18
Gain > Hiero (1000-best) + Nmt	+0.84	+0.59	+0.39	+0.74	+0.45

NMT model. On top of the stronger baseline, *Hiero (1000-best)* + Nmt, gappy composition still yields an improvement of up to ~ 0.9 Bleu and Meteor.

Table 6 shows the results broken down by gap types. In comparison with the Hiero baseline, we see a slightly different picture than for English–German, with the End and Brace gap types yielding the largest improvements. However, the improvements for all gap types are above 1 Bleu which indicates that the English–French language pair is less sensitive to the position of source gaps. After 1000-best rescoring of full hypotheses with the NMT system, the remaining improvements are largest (+0.84 Bleu) for the Middle gap type, as for English–German. This provides further evidence that translation systems are more likely to benefit from this gap type, which removes discontinuities in the input.

6.3. *Reachability of skeleton translations*

Translating source skeletons with the Hiero system can produce translations which are not substrings of any translations in the full translation lattices, or receive very low scores and are likely to be pruned. For this reason, composition with an n-best list of skeleton translations can fail, in which case we fall back to regular Hiero decoding. Table 7 shows the percentage of failed compositions when performing gappy composition with a 1-best or 1000-best list of skeleton hypotheses. While we see a fairly high failure rate of ~ 16% to ~ 21% for 1-best skeletons for English–German and ~ 9% to ~ 13% for English–French, we can reduce it to ~ 1% or less when composing with 1000-best lists. This shows that on the one hand, considering a large space of skeleton hypotheses is important and on the other hand that our approach could potentially be improved by extending our Hiero system such that more high-quality skeleton hypotheses can be reached. Another option to increase the influence of skeleton translations would be to allow for token mismatches at a certain cost, such that composition with a skeleton could succeed even if not all of its tokens match a given full hypothesis.

6.4. *Impact of NMT on translations*

The advantage of combining Hiero with an NMT system is that we can benefit from the strengths of both systems. For example, Hiero ensures that all source words are covered during translation, while neural MT systems with attention mechanism can suffer from over-translation and under-translation, as pointed out by Tu et al. (2016). On the

Table 7
Percentage of failed compositions between Hiero and the n-best skeleton translations under the NMT model with gappy composition.

Set	English–German		English–French	
	n=1 (%)	n=1000 (%)	n=1 (%)	n=1000 (%)
newstest2012-subset	20.6	0.6	13.2	1.2
newstest2013-subset	16.1	0.6	9.0	0.8
newstest2014-subset	15.8	0.6	10.6	0.4

other hand, the NMT system is better at using contextual information on the source side by relying on recurrent forward and backward source annotations. Using source skeletons can have a direct impact on these source annotations. For example, if we remove a relative clause from the source sentence, this is likely to result in more informative source annotations for tokens before and after the relative clause.

7. Discussion and future work

Our current approaches to decoding with original and simplified inputs have allowed us to gain a better understanding of the potential of such additional information for translation. However, several open questions remain which we would like to address in the future.

7.1. Language-dependent issues with simplification

Inspecting the annotated source skeletons and translation outputs reveals that there is a certain language-specific component to what constitutes a useful or harmful simplification. For example, consider the multi-word unit *members of parliament* in the first example of Fig. 1. When the target language is German, this could be translated as a single word such as *Abgeordneter*. Therefore, deleting the prepositional phrase *of parliament* could mean that lexical choice in the translation of the skeleton would differ from lexical choice in the translation of the full sentence. This suggests that it could be worth investigating a bilingual approach to source simplification that takes the target language into account.

Further, in languages where word order differs depending on the type of clause (main, subordinate etc.), it can potentially be harmful to produce a sentence skeleton which turns a subordinate clause into a main clause. Even though considering a large *n*-best list of skeleton translations mitigates the effects of word order, a reasonable strategy could be to use only certain types of source skeletons and again, this may depend on the language pair and in particular the target language.

In this work, we have focused on English as a source language and have varied the target language. One practical reason for this was our need for manual annotations and the ease of recruiting English-speaking annotators. It would be interesting to compare the effect of source simplification in the opposite translation directions where German and French inputs would undergo simplification. Previous research has been limited to English and Chinese as source languages with systems for English−Spanish (Mellebeek et al., 2006), English−Japanese (Sudoh et al., 2010) and Chinese−English (Xiao et al., 2014). Given its syntactic structure where verb complexes are often separated by their arguments, German could be a very promising language for source simplification. However, without experimental evidence it is difficult to predict which pairs of source and target languages would be most suitable for simplification.

7.2. Integration of automatic simplification methods

A reasonable next step in this line of work is to integrate automatic simplification methods to replace the gold annotations currently used by our models. As mentioned previously, there exist several approaches to tackle this problem by deleting words from a given input sentence using a discriminative model (McDonald, 2006; Cohn and Lapata, 2007), transforming the input using a set of transformation rules (Cohn and Lapata, 2008) or using LSTMs to predict which words from the input to keep (Filippova et al., 2015).

We have carried out initial, unreported experiments with automatic simplification based on dependency parsing with subtree and phrase deletion. While this approach yields good simplifications in many cases, we show some problematic ones in Fig. 6. In the first example, the main verb *reply* was mistagged and *had* is instead assumed to be the main verb. In the second example, *thanks* was erroneously marked as subject of *pay* in addition to *neighbours* and is therefore retained in the output. In the third example, *through* is marked as an adverbial of *thought* and therefore it is not clear whether it should be kept or removed. Similarly, we may not want to remove the prepositional phrase *of notebook* in the last example sentence while in other instances removing a prepositional phrase would be unproblematic. We also experimented with the *t3* toolkit described in Cohn and Lapata (2008). However, we found the output of *t3* on this data set to be less grammatical than the output of our own simplifier and therefore did not proceed to use these simplifications for translation. One reason for this could be that it was developed for simplification as a standalone task and not with translation in mind.

1. the opponents of the current president reply **that Reagan** also **had simple ideas** and that he won the cold war **.**
2. **on the other hand , thanks** to the constancy of energy policy , **our neighbours pay** 40 % more for their electricity than French households **.**
3. **this ambivalence must be thought** *through* , in order that it can be bypassed **.**
4. **this type** *of notebook* **is said to be highly prized by writers** and travellers **.**

Fig. 6. Examples of automatic input simplification using a dependency-based method. Examples 1. and 2. have tagging/parsing errors while in examples 3. and 4. the decision which tokens to keep is difficult without additional information about the tokens involved, as indicated in italics.

Due to these issues, our attempts at using automatic simplifications for machine translation have been less successful in improving performance so far. We intend to investigate these issues further in future work, with a particular focus on neural models for simplification.

7.3. Integrating source simplification with neural machine translation

Another avenue for future work could be to integrate skeleton information directly into an NMT system, either to guide the attention mechanism when translating the full source sentence or to construct better source annotations by ignoring gaps when building the recurrent source representations. Since the simplification problem can also be approached with neural models (Filippova et al., 2015), it may even be possible to design a neural model that performs source simplification as part of the translation process.

8. Conclusions

We have investigated the potential of source sentence simplification to provide additional inputs for machine translation systems. We have shown two decoding approaches that can use skeleton inputs with the goal of implicitly improving syntactic well-formedness and fluency for complex sentences and found that in most cases, not limiting the derivations to be consistent with source skeletons leads to better performance. Using the Hiero system to score full and skeleton hypotheses only resulted in small performance improvements, but scoring the skeleton translations with a more powerful model (a combination of Hiero and neural machine translation) confirmed that skeleton information can be useful for translation. We show small but consistent gains for English−German on a strong baseline that includes rescoring the full hypotheses with a neural translation model and larger gains for English−French. Further analysis shows that the type of skeleton resulting from simplification is an important factor for the success of skeleton-based translation. For English−German, our gappy composition approach improves by ~ 0.8 Bleu over the strong baseline when simplification removes tokens from the middle of the sentence. For English−French, we see improvements for all skeleton types which indicates that this language pair is less sensitive to the position of gaps. Still, in comparison to the stronger baseline, the improvements remain largest when removing tokens from the middle of the sentence. We release our annotated data to the community to encourage further research on source simplification for machine translation.

In the future, we plan to further experiment with automatic simplification as well as improving our decoding procedures to account for a larger space of skeleton hypotheses. The use of simplified inputs for translation is not limited to any particular translation framework and could potentially be integrated into a neural translation model, for example by modifying the attention mechanism or the construction of source annotations.

Acknowledgment

This work was supported by the EPSRC grant Improving Target Language Fluency in Statistical Machine Translation, grant number EP/L027623/1.

References

Bahdanau, D., Cho, K., Bengio, Y., 2015. Neural machine translation by jointly learning to align and translate. In: Proceedings of the International Conference on Learning Representations, pp. 1–15.

Banerjee, S., Lavie, A., 2005. METEOR : An automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the Association for Computational Linguistics Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization.

Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Huck, M., Hokamp, C., Koehn, P., Logacheva, V., Monz, C., Negri, M., Post, M., Scarton, C., Specia, L., Turchi, M., 2015. Findings of the 2015 workshop on statistical machine translation. In: Proceedings of the 10th Workshop on Statistical Machine Translation.

Braune, F., Gojun, A., Fraser, A., 2012. Long-distance reordering during search for hierarchical phrase-based SMT. In: Proceedings of the European Association for Machine Translation-2012 (May), pp. 28–30.

Chiang, D., 2005. A hierarchical phrase-based model for statistical machine translation. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics 2005. Morristown, NJ, USA, pp. 263–270.

Chiang, D., 2007. Hierarchical phrase-based translation. In: Proceedings of the Computational Linguistics, 33, pp. 201–228. (2)

Chiang, D., DeNeefe, S., Pust, M., 2011. Two easy improvements to lexical weighting. In: Proceedings of the Association for Computational Linguistics: Human Language Technologies.

Cohn, T., Lapata, M., 2007. Large margin synchronous generation and its application to sentence compression. In: Proceedings of the Empirical Methods on Natural Language Processing and Computational Natural Language Learning, pp. 73–82.

Cohn, T., Lapata, M., 2008. Sentence compression beyond word deletion. In: Proceedings of the Computational Linguistics.

Deng, Y., Byrne, W., 2006. An alignment toolkit for statistical machine translation. In: Proceedings of the Human Language Technologies-North American Chapter of the Association for Computational Linguistics Demonstrations Program, pp. 1–4.

Filippova, K., Alfonseca, E., Colmenares, C.A., Kaiser, L., Vinyals, O., 2015. Sentence compression by deletion with LSTMs. In: Proceedings of the Empirical Methods in Natural Language Processing, pp. 360–368. (September)

de Gispert, A., Iglesias, G., Blackwood, G., Banga, E.R., Byrne, W., 2010. Hierarchical phrase-based translation with weighted finite-state transducers and shallow-n grammars. Comput.Linguist. 36 (3), 505–533. (October)

Heafield, K., Pouzyrevsky, I., Clark, J.H., Koehn, P., 2013. Scalable modified Kneser−Ney language model estimation. In: Proceedings of the Association for Computational Linguistics, pp. 690–696.

Iglesias, G., De Gispert, A., Banga, E., Byrne, W., 2009. Hierarchical phrase-based translation with weighted finite state transducers. In: Proceedings of the Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (June), pp. 433–441.

Jean, S., Cho, K., Memisevic, R., Bengio, Y., 2015. On using very large target vocabulary for neural machine 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, pp. 1–10.

Macherey, W., Och, F.J., Thayer, I., Uszkoreit, J., 2008. Lattice-based minimum error rate training for statistical machine translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Morristown, NJ, USA, p. 725.

McDonald, R.T., 2006. Discriminative sentence compression with soft syntactic evidence. In: Proceedings of European Chapter of the Association for Computational Linguistics, pp. 297–304.

Mellebeek, B., Owczarzak, K., Groves, D., van Genabith, J., Way, A., 2006. A syntactic skeleton for statistical machine translation. In: Proceedings of the 11th Conference of the European Association for Machine Translation).

van Merrienboer, B., Bahdanau, D., Dumoulin, V., Serdyuk, D., Warde-farley, D., Chorowski, J., Bengio, Y., 2015. Blocks and Fuel : Frameworks for deep learning. arXiv preprint arXiv:1506.00619

Mohri, M., 2003. Learning from uncertain data. In: Proceedings of The 16th Annual Conference on Computational Learning Theory (COLT 2003).

Och, F., Ney, H., 2001. Statistical multi-source translation. In: Proceedings of the Machine Translation Summit VIII.

Papineni, K., Roukos, S., Ward, T., Zhu, W.-J., 2002. BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of Association for Computational Linguistics, pp. 311–318.

Pauls, A., Klein, D., 2012. Large-scale syntactic language modeling with treelets. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (July), pp. 959–968.

Pouget-Abadie, J., Bahdanau, D., van Merrienboer, B., Cho, K., Bengio, Y., 2014. Overcoming the curse of sentence length for neural machine translation using automatic segmentation. In: Proceedings of the Syntax, Semantics and Structure in Statistical Translation 2014, pp. 78–85.

Schroeder, J., Cohn, T., Koehn, P., 2009. Word lattices for multi-source translation. In: Proceedings of the European Chapter of the Association for Computational Linguistics 2009 (April), pp. 719–727.

Schwenk, H., 2014. http://www-lium.univ-lemans.fr/~schwenk/cslm_joint_paper/README.

Schwenk, H., Koehn, P., 2008. Large and diverse language models for statistical machine translation. In: Proceedings of the International Joint Conference on Natural Language Processing, pp. 661–666.

Sennrich, R., 2015. Modelling and optimizing on syntactic N-grams for statistical machine translation. In: Proceedings of the Transactions of the Association for Computational Linguistics, 3, pp. 169–182.

Shen, L., Xu, J., Weischedel, R., 2008. A new string-to-dependency machine translation algorithm with a target dependency language Model. In: Proceedings of the Association for Computational Linguistics-08: Human Language Technologies, pp. 577–585.

Stahlberg, F., Hasler, E., Waite, A., Byrne, B., 2016. Syntactically guided neural machine translation. In: Proceedings of the Association for Computational Linguistics, pp. 299–305.

Stolcke, A., 2002. SRILM − An extensible language modeling toolkit. In: Proceedings of the International Conference on Spoken Language Processing, vol. 2, pp. 901–904.

Sudoh, K., Duh, K., Tsukada, H., Hirao, T., Nagata, M., 2010. Divide and translate : Improving long distance reordering in statistical machine translation. In: Proceedings of the Joint 5th Workshop on Statistical Machine Translation and Metrics, pp. 418–427.

Tu, Z., Lu, Z., Liu, Y., Liu, X., Li, H., 2016. Modeling coverage for neural machine translation. arXiv preprint arXiv:1601.04811

Xiao, T., Zhu, J., Zhang, C., 2014. A hybrid approach to skeleton-based translation. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 563–568.

Xiong, H., Xu, W., Mi, H., Liu, Y., Liu, Q., 2009. Sub-sentence division for tree-based machine translation. In: Proceedings of the Association for Computational Linguistics and Asian Federation of Natural Language Processing 2009 Conference Short Papers (August), pp. 137–140.

Zoph, B., Knight, K., 2016. Multi-source neural translation. arXiv preprint arXiv:1601.00710

Please cite this article as: E. Hasler et al., Source sentence simplification for statistical machine translation, Computer Speech & Language (2017), http://dx.doi.org/10.1016/j.csl.2016.12.001