Semantic Structural Decomposition for Neural Machine Translation

Elior Sulem[*]
Dept. of Computer and Information Science
University of Pennsylvania
eliors@seas.upenn.edu

Omri Abend **Ari Rappoport**
Dept. of Computer Science
The Hebrew University of Jerusalem
oabend|arir@cs.huji.ac.il

Abstract

Building on recent advances in semantic parsing and text simplification, we investigate the use of semantic splitting of the source sentence as preprocessing for machine translation. We experiment with a Transformer model and evaluate using large-scale crowd-sourcing experiments. Results show a significant increase in fluency on long sentences on an English-to-French setting with a training corpus of 5M sentence pairs, while retaining comparable adequacy. We also perform a manual analysis which explores the tradeoff between adequacy and fluency in the case where all sentence lengths are considered.[1]

1 Introduction

In this paper, we apply a semantic decomposition approach for Neural Machine Translation (NMT) and demonstrate that it can tackle two of the main limitations of state-of-the-art NMT. The first is the translation of long sentences, which is a recurrent issue arising in NMT evaluation (Sutskever et al., 2014; Cho et al., 2014; Pouget-Abadie et al., 2014; Su et al., 2018; Currey and Heafield, 2018). The second limitation is that current research in NMT mostly focuses on translating single sentences to single sentences, and is evaluated accordingly. However, Li and Nenkova (2015) showed that using several sentences to translate a source sentence is sometimes the preferable option. Therefore, the simplicity of the output could be an important quality marker for translation.

In our model, each source sentence is split (or decomposed) into semantic units, namely scenes, building on the Direct Semantic Splitting algorithm (DSS; Sulem et al., 2018b) that uses the Universal Conceptual Cognitive Annotation (UCCA; Abend and Rappoport, 2013) scheme for semantic representation. Scenes are then translated separately and concatenated for generating the final translation output, which may consist of several sentences.

Our main experiments use the state-of-the-art Transformer model (Vaswani et al., 2017) in English-to-French settings. We also include experiments with other MT architectures and training set sizes, and evaluate our results using the crowd-sourcing protocol of Graham et al. (2016) (§4). We obtain a significant increase in fluency on sentences longer than 30 words on the newstest2014 test corpus for English-to-French translation, with a training corpus of 5M sentence pairs, without degrading adequacy. Considering all sentence lengths, we observe a tradeoff between fluency and adequacy. We explore it using a manual analysis, suggesting that the decrease in adequacy is partly due to the loss of cohesion resulting from the splitting (§6).

We then proceed to investigate the case of simulated low-resource settings as well as the effect of other sentence splitting methods, including Split-and-Rephrase models (Aharoni and Goldberg, 2018; Botha et al., 2018) (§7). The latter yield considerably lower scores than the use of simple semantic rules, supporting the case for corpus-independent simplification rules.

2 Related Work

Sentence segmentation for MT. Segmenting sentences into sub-units, based on punctuation and syntactic structures, and recombining their output has been explored by a number of statistical MT works (Xiong et al., 2009; Goh and Sumita, 2011; Sudoh et al., 2010). In NMT, Pouget-Abadie et al. (2014) segmented the source using ILP, tackling English-to-French neural translation. They con-

[*] This work was done when being affiliated to the Hebrew University of Jerusalem

[1] The code and the evaluation data are available at https://github.com/eliorsulem/Semantic-Structural-Decomposition-for-NMT

cluded that segmentation improves overall translation quality but quality may decrease if the segmented fragments are not well-formed. The concatenation may sometimes degrade fluency and result in errors in punctuation and capitalization. Kuang and Xiong (2016) attempted to find split positions such that no reordering will be necessary in the target side for Chinese-English. We differ from these approaches in using a separate text simplification module that can be applied to different kinds of MT systems, and using a semantically-motivated segmentation. Moreover, we allow the final output to be composed of several sentences, taking into account the structural simplicity aspect of translation quality (Li and Nenkova, 2015).

Text Simplification for MT. Sentence splitting, which goes beyond segmentation and denotes the conversion of one sentence into one or several sentences, is the main structural operation studied in Text Simplification (TS). While MT preprocessing was one of the main motivations for the first automatic simplification system (Chandrasekar et al., 1996), only few works empirically explored the usefulness of simplification techniques for MT.

Mishra et al. (2014) used sentence splitting as a preprocessing step for Hindi-to-English translation with a dependency parser and additional modules for gerunds and shared arguments. Štajner and Popović (2016) performed structural and lexical simplification as part of a preprocessing step for English-to-Serbian MT. Manual correction is carried out before translation. Štajner and Popović (2018) investigated the use of TS as a processing step for NMT, focusing on syntax-based rules that address relative clauses (Siddhathan, 2011) for English-to-German and English-to-Serbian translation. Investigating the translation of 106 out of 1000 sentences that have been modified by simplification, they find that the automatic simplification of English relative clauses can improve translation only if simplifications are quality-controlled or corrected in post-processing. We differ from this work in using semantic rules and by translating independently each of the obtained sentences.

3 Semantic Decomposition

UCCA (Universal Cognitive Conceptual Annotation; Abend and Rappoport, 2013) is a semantic annotation scheme rooted in typological and cognitive linguistic theory (Dixon, 2010b,a; Langacker, 2008). It aims to represent the main semantic phenomena in the text, abstracting away from syntax.

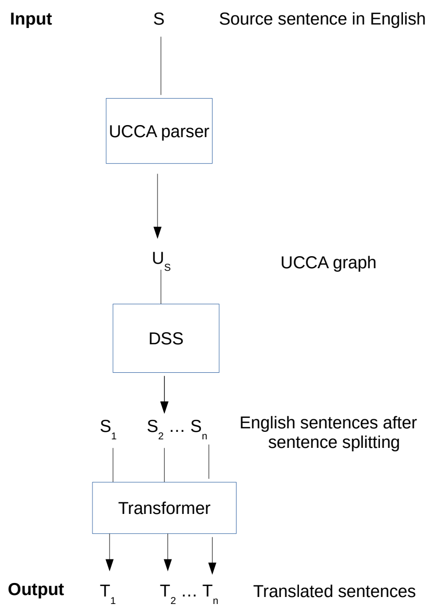

Figure 1: SemSplit pipeline. After the application of the Direct Semantic Splitting, which requires UCCA parsing, each of resulted sentences is independently translated using a Transformer system, previously trained on English-French parallel data. The obtained translations are directly concatenated, forming the final output.

Formally, UCCA structures are directed acyclic graphs whose nodes (or *units*) correspond either to the leaves of the graph or to several elements viewed as a single entity according to some semantic or cognitive consideration. A *scene* is UCCA's notion of an event or a frame, and is a unit that corresponds to a movement, an action or a state which persists in time. Every scene contains one main relation, which can be either a Process or a State. Scenes may contain one or more Participants, interpreted in a broad sense to include locations and destinations. For example, the sentence "John went home" has a single scene whose Process is "went". The two Participants are "John" and "home".

Scenes can provide additional information about an established entity (Elaborator scenes), commonly participles or relative clauses. For example, "(child) who went home" is an Elaborator scene in "The child who went home is John". A scene may also be a Participant in another scene. For example, "John went home" in the sentence: "He said John went home". In other cases, scenes are annotated as parallel scenes (H), which are flat structures and may include a Linker (L), as in: "When$_L$ [he arrives]$_H$, [he will call them]$_H$".

For UCCA parsing, we use TUPA, a transition-based parser (Hershcovich et al., 2017) (specifically, the TUPA$_{BiLSTM}$ model).

We build on the DSS rule-based semantic splitting method (Sulem et al., 2018b), and use Rule #1 which targets parallel scenes. We further explore the use of the additional kinds of scenes in Section 7 for less conservative sentence splitting. In Rule #1, parallel scenes of a given sentence are extracted, split into different sentences and concatenated according to the order of appearance. More formally, given a decomposition of a sentence S into parallel scenes $Sc_1, Sc_2, \cdots Sc_n$ (indexed by the order of the first token), we obtain the following rule, where "|" is the sentence delimiter:

$$S \longrightarrow Sc_1|Sc_2|\cdots|Sc_n$$

As UCCA allows argument sharing between scenes, the rule may duplicate the same sub-span of S across sentences. For example, the rule will convert "He came back home and played the piano" into "He came back home"|"He played the piano.".

Using UCCA-based sentence splitting in our model is motivated by the corpus-based analysis presented in Sulem et al. (2015) where it is shown that a scene in English is generally translated to a scene in French.

4 Experimental Setup

Corpora We experiment on the full English-French training data provided in the WMT setting (Bojar et al., 2014), which corresponds to about 39M sentence pairs after cleaning.[2] We refer to this setting as the FullTrain Setting. We also experiment on the LessTrain Setting where less training data is involved by removing the large UN Corpus and the 10^9 French-English Corpus from the training data, obtaining a new training corpus of about 5M sentence pairs. The development set is Newstest 2013, that consists of 3000 sentences. The test set is Newstest2014, consisting of 3003 sentences.

Systems To investigate the use of semantic structural decomposition for NMT, we propose a two-step method. First, the original sentence is split into several sentences the DSS rule (see § 3), implemented with the UCCA software.[3]. Then, each of the obtained sentences is translated separately by the OpenNMT-py implementation of the Transformer (Vaswani et al., 2017).[4] The translated sentences are concatenated to form the final output. We name the combined system Transformer SemSplit and compare it to the Transformer Baseline, where no splitting is performed. The pipeline architecture is summarized in Figure 1.

The Transformer is trained for 200K training steps, both in the FullTrain and the LessTrain settings. The development data was used for selecting the model with the highest accuracy (where perplexity was used in cases of ties). The system was evaluated on the development data every 10K steps.

For comparison, we also implement our system in the case where the Transformer is replaced by another NMT system, namely a two-layers LSTM model and the Moses phrase-based machine translation system (Koehn et al., 2007). The neural model, also implemented with OpenNMT-py, is trained and validated in the same way as the Transformer. For Moses, the default model is used in a single setting (LessTrain) with MGIZA word alignment,[5] and KenLM language model (Heafield, 2011) using the monolingual data provided in WMT 2014, and MERT tuning on the development set. Here too we compare the combined systems to baseline systems which do not perform decomposition.

4.1 Evaluation Using Crowdsourcing

In addition to the limitations of BLEU evaluation (Papineni et al., 2002) in the context of MT (Callison-Burch et al., 2006, and much subsequent work), BLEU may correlate negatively with output quality in cases that involve sentence splitting (Sulem et al., 2018a). We therefore evaluate using crowdsourcing, and follow the protocol proposed by Graham et al. (2016). Evaluation was carried out using Amazon Mechanical Turk.[6] See Appendix A for a detailed description.

5 Results

The results in both FullTrain and LessTrain settings are presented in Table 1.

In terms of fluency, LessTrain Transformer SemSplit ranks first in this setting and significantly outperforms the corresponding baseline system (52.5 vs. 42.5, $p < 10^{-4}$).[7] For Moses too, the use

[2] Cleaning, tokenization and truecasing as well as detokenization and detruecasing of the outputs are performed using the Moses tools: http://www.statmt.org/moses/.

[3] https://github.com/danielhers/ucca

[4] https://github.com/OpenNMT/OpenNMT-py

[5] https://github.com/moses-smt/mgiza

[6] https://www.mturk.com/

[7] Significance is computed using the Wilcoxon one-sided rank sum test applied on the standardized scores, following Graham et al. (2016).

System		Adequacy		Fluency	
		All	Long	All	Long
Transformer	Baseline	48.8	48.0	57.1	49.4
	SemSplit	40.0	28.7	43.5	37.1
LSTM	Baseline	41.2	50.1	43.2	46.8
	SemSplit	33.1	34.7	39.3	37.5
Transformer	Baseline	47.5	41.7	42.5	39.6
	SemSplit	39.8	40.1	52.5	52.1
LSTM	Baseline	40.6	37.5	47.3	52.9
	SemSplit	36.8	34.9	46.6	44.5
Moses	Baseline	40.1	45.4	38.1	30.1
	SemSplit	34.9	43.2	40.2	50.4

Table 1: Raw system adequacy and fluency scores of the SemSplit systems and the baselines on the FullTrain (top) and LessTrain (bottom) settings. For each system, the raw score is presented, both when considering every sentence length (**All**) and when focusing on sentences longer than 30 words (**Long**).

System		Adequacy	Fluency
Transformer	Baseline	47.0	47.5
	SemSplit$_{1+2}$	29.7	32.5
	NeuralWiki-Split	12.9	6.0
	NeuralWEB-SPLIT	4.1	5.1
LSTM	Baseline	40.6	38.2
	SemSplit$_{1+2}$	24.3	34.2
Transformer	Baseline	50.6	55.1
	SemSplit$_{1+2}$	25.3	39.1
LSTM	Baseline	45.6	46.4
	SemSplit$_{1+2}$	31.9	31.0
Moses	Baseline	38.2	38.3
	SemSplit	30.8	23.5

Table 2: Raw system adequacy and fluency scores of the SemSplit$_{1+2}$ systems and the baselines on the FullTrain (top) and LessTrain (bottom) setting.

of semantic sentence splitting increases fluency (40.2 vs. 38.1), but not significantly. On the other hand, where splitting is used as preprocessing, adequacy scores decrease. In particular, LessTrain Transformer Baseline significantly outperforms the SemSplit counterpart (47.5 vs. 39.8, $p < 10^{-4}$).

For sentences longer than 30, SemSplit Transformer in the LessTrain setting significantly outperforms the baseline in terms of fluency (52.1 vs. 39.6, $p = 0.02$), with only a non-significant (small) degradation in adequacy (41.7 vs. 40.1, $p = 0.46$).

6 Manual Analysis

To further zoom in on the obtained adequacy scores, we decompose adequacy into two dimensions: preservation of semantic content in the level of scenes and the cohesion of the text (i.e., whether the different scenes are cohesively linked together). To do so, we manually annotate a sample of 150 sentences from the original test set with a similar proportion of sentences in different length categories as the original corpus, and assess the semantic preservation at the scene-level for each of the extracted scenes, as well as the sentence-level cohesion (see Appendix B for the protocol).

For LessTrain, we find that 66.2% of the scenes are deemed equally preserved by the SemSplit and Baseline systems. On the other hand, 20.9% of the scenes are better preserved by the baseline and 10.7% of the scenes are better preserved by the SemSplit system. Averaging over scenes that belong to the same sentence, we find that 68% of the sentences are either better preserved by SemSplit or equally preserved. Regarding cohesion, SemSplit and the Baseline have a comparable cohesion for 59% of the sentences. The Baseline has a better cohesion for 36% of the sentences, while it is improved by SemSplit in 5% of the cases.

The analysis suggests that cohesion has a central role in the decrease (and the non-increase for long sentences) of the adequacy scores. Therefore the tradeoff between adequacy and fluency observed when all sentence lengths are considered can be explained by a tradeoff between the cohesion and structural simplicity aspects of translation quality.

The different aspects of the translation quality are further illustrated in Table 3, where two input and output examples are presented, focusing on the LessTrain setting. In example (1), the SemSplit output is similar to the Baseline one at the lexical level but differs in its structure, the SemSplit system behaving as a cross-lingual simplifier at the structural level. On the other hand, linkers such as ”so” are not translated in the case of SemSplit. In example (2), the word ”interference” is correctly translated by SemSplit, while it is translated into ”ingérence” (”intervention”) in French, which is wrong in this context.

7 Additional Experiments

We first explore the performance of the proposed system in low-resource machine translation, by following the approach of Hoang et al. (2018) and randomly select 1M and 100K sentence pairs from the entire English-French training set, defining the 1MTrain and 100KTrain settings respectively. Tuning and testing remain as before.

The resulted raw scores for the 1MTrain and 100KTrain settings are presented in Appendix D, Table 4 . We observe that while in 1MTrain, the SemSplit models obtain low results compared to the respective baselines, the SemSplit models obtain higher fluency in 100KTrain, though not significantly.

Second, to further explore the sentence splitting component, we replicate our model, separating both parallel and embedded scenes before the

(1) **Input:**	Hamas has defended its use of tunnels in the fight against Israel, stating that the aim was to capture Israeli soldiers so they could be exchanged for Palestinian prisoners.
Baseline (LessTrain)	**Output:** Le Hamas a défendu son utilisation de tunnels dans la lutte contre Israël, affirmant que l'objectif était de capturer des soldats israéliens afin qu'ils puissent être échangés contre des prisonniers palestiniens.
	Literal translation: Hamas has defended its use of tunnels in the fight against Israel, stating that the aim was to capture Israeli soldiers so they could be exchanged for Palestinian prisoners.
SemSplit (LessTrain)	**Output:** Le Hamas a défendu son utilisation de tunnels dans la lutte contre Israël. Le Hamas a déclaré que l'objectif était de capturer des soldats israéliens. Ils pourraient être échangés contre des prisonniers palestiniens.
	Literal translation: Hamas has defended its use of tunnels in the fight against Israel. Hamas stated that the aim was to capture Israeli soldiers. They could be exchanged for Palestinian prisoners.
(2) **Input:**	Douglas Kidd of the National Association of Airline Passengers said he believes interference from the devices is genuine even if the risk is minimal.
Baseline (LessTrain)	**Output:** Douglas., de l'Association nationale des compagnies aériennes, a déclaré qu'il considérait que l'ingérence des appareils était réelle, même si le risque était minimal.
	Literal translation: Douglas., from the Association National of the companies airline, claimed that he believed that the intervention of the devices was genuine, even if the risk is minimal.
SemSplit (LessTrain)	**Output**: Douglas., de l'Association nationale des compagnies aériennes, a déclaré qu' il estimait que l'interférence avec les appareils était réelle. Le risque est minimal.
	Literal translation: Douglas., from the Association national of the companies airline, claimed that he believed the interference with the devices was genuine. The risk is minimal.

Table 3: Input and output examples for the Baseline and SemSplit system in the LessTrain setting, together with an English literal translation of the French outputs.

translation. We use Rule #2 from the DSS system (Sulem et al., 2018b) addressing Elaborator scenes (See Appendix C), which we further extend to also include Participant scenes. We denote the resulting system with Transformer SemSplit$_{1+2}$. We also compare the model with two additional sentence splitting systems, where DSS is replaced with the Seq2Seq Copy 512 model for Split-and-Rephrase (Aharoni and Goldberg, 2018) trained on the WEB-SPLIT corpus (Narayan et al., 2017) (version 1.0), and the same model trained on the WikiSplit corpus (Botha et al., 2018). Each of the obtained new sentences is translated by the FullTrain Transformer system. Finally the translated sentences are directly concatenated. The resulting systems are denoted with Transformer NeuralWEB-SPLIT and Tranformer NeuralWiki-Split.

The results for the FullTrain and LessTrain settings are presented in Table 2. As in the case where only the first rule is used, adequacy scores decrease following splitting. On the other hand, in this case the SemSplit models do not have higher fluency scores than their corresponding baselines, probably because of the more aggressive splitting compared to #Rule 1 alone. For both adequacy and fluency, the Split-and-Rephrase models obtain very low scores. Observing their outputs, we find many wrong splits and word repetitions at the splitting phase, which affects the final output. As this trend is not observed on the standard WEB-SPLIT test corpus, these results may suggest a domain adaptation effect, which supports the case for corpus-independent sentence splitting.

8 Conclusion

This work investigates the application of semantic structural decomposition for NMT, proposing an intermediate way between sentence segmentation used in MT and TS preprocessing, where each of the semantic components is separately translated. Using the Transformer and large-scale crowd-sourcing evaluation, we obtain an increase in fluency on long sentences on an English-to-French setting without significantly lowering adequacy. We further observe increased fluency when evaluating on all the sentences, albeit at the cost of adequacy. Future work concerns the recombination of the output sentences, inserting the linkage between them, so as not to lose semantic content.

Aknowledgments

We would like to thank the annotators for participating in our evaluation experiments and in the UCCA annotation. This work was partially supported by the Israel Science Foundation (grant No.929/17) and by the HUJI Cyber Security Research Center in conjunction with the Israel National Cyber Bureau in the Prime Minister's Office.

References

Omri Abend and Ari Rappoport. 2013. Universal Conceptual Cognitive Annotation (UCCA). In *Proc. of ACL-13*, pages 228–238.

Omri Abend, Shai Yerushalmi, and Ari Rappoport. 2017. UCCAApp: Web-application for syntactic and semantic phrase-based annotation. In *Proc. of ACL'17, System Demonstrations*, pages 109–114.

Roee Aharoni and Yoav Goldberg. 2018. Split and rephrase: Better evaluation and a stronger baseline. In *Proc. of ACL'18 (Short papers)*, pages 719–728.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Jan A. Botha, Manaal Faruqui, John Alex, Jason Baldridge, and Dipanjan Das. 2018. Learning to split and rephrase from Wikipedia edit history. In *Proc. of EMNLP'18 (Short papers)*, pages 732–737.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of BLEU in machine translation. In *Proc. of EACL'06*, pages 249–256.

Raman Chandrasekar, Christine Doran, and Bangalore Srinivas. 1996. Motivations and methods for sentence simplification. In *Proc. of COLING'96*, pages 1041–1044.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proc. of Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*.

Anna Currey and Kenneth Heafield. 2018. Multi-source syntactic neural machine translation. In *Proc. of EMNLP'18*, pages 2961–2966.

Robert M.W. Dixon. 2010a. *Basic Linguistic Theory: Grammatical Topics*, volume 2. Oxford University Press.

Robert M.W. Dixon. 2010b. *Basic Linguistic Theory: Methodology*, volume 1. Oxford University Press.

Chooi-Ling Goh and Eiichiro Sumita. 2011. Splitting long input sentences for phrase-based statistical machine translation. In *Proc. of ANLP'11*, pages 802–805.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2016. Can machine translation be evaluated by the crowd alone? *Natural Language Engineering*, 1(1):1–28.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proc. of the Sixth Workshop on Statistical Machine Translation*.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. A transition-based directed acyclic graph parser for UCCA. In *Proc. of ACL'17*, pages 1127–1138.

Cong Duy Vu Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proc. of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Buch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proc. of ACL'07 on interactive poster and demonstration sessions*, pages 177–180.

Shaoui Kuang and Deyi Xiong. 2016. Automatic long sentence segmentation for neural machine translation. In *Natural Language Understanding and Intelligent Applications: 5th CCF Conference on Natural Language Processing and Chinese Computing, NLPCC 2016, and 24th International Conference on Computer Processing of Oriental Languages, ICCPOL 2016*, pages 162–174.

Ronald W. Langacker. 2008. *Cognitive Grammar: A Basic Introduction*. Oxford University Press, USA.

Junyi Jessi Li and Ani Nenkova. 2015. Detecting content-heavy sentences: A cross-language case study. In *Proc. of EMNLP'15*, pages 1271–1281.

Kshitij Mishra, Ankush Soni, Rahul Sharma, and Dipti Misra Sharma. 2014. Exploring the effects of sentence simplification on Hindi to English Machine Translation systems. In *Proc. of the Workshop on Automatic Text Simplification: Methods and Applications in the Multilingual Society*, pages 21–29.

Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. 2017. Split and rephrase. In *Proc. of EMNLP'17*, pages 617–627.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. of ACL'02*, pages 311–318.

Jean Pouget-Abadie, Dzmitry Bahdanau, Bart van Merriënboer, Kyunghyun Cho, and Yoshua Bengio. 2014. Overcoming the curse of sentence length for neural machine translation with automatic segmentation. In *Proc. of the Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 78–85.

Advaith Siddhathan. 2011. Text simplification using typed dependencies: A comparison of the robustness of different generation strategies. In *Proc. of the 13th European Workshop on Natural Language Generation*, pages 2–11. Association of Computational Linguistics.

Sanja Štajner and Maja Popović. 2016. Can text simplification help machine translation. *Baltic J. Modern Computing*, 4:230–242.

Jinsong Su, Jiali Zeng, Deyi Xiong, Yang Liu, Mingxuan Wang, and Jun Xie. 2018. A hierarchy-to-sequence attentional neural machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(3).

Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, Tsutomu Hirao, and Masaaki Nagata. 2010. Divide and translate: Improving long distance reordering in statisticxal machine translation. In *Proc. of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR*, pages 418–427.

Elior Sulem, Omri Abend, and Ari Rappoport. 2015. Conceptual annotations preserve structure across translations. In *Proc. of 1st Workshop on Semantics-Driven Statistical Machine Translation (S2Mt 2015)*, pages 11–22.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018a. BLEU is not suitable for the evaluation of text simplification. In *Proc. of EMNLP*, pages 738–744.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018b. Simple and effective text simplification using semantic and neural methods. In *Proc. of ACL*, pages 162–173.

Ilya Sutskever, Oriol Vinyals, and Quôc Lê. 2014. Sequence to sequence learning with neural networks. In *Proc. of NIPS*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. of NIPS*.

Sanja Štajner and Maja Popović. 2018. Improving machine translation of English relative clauses with automatic text simplification. In *Proc. of the INLG 2018 First Workshop on Automatic Text Adaptation (ATA)*.

Hao Xiong, Wenwen Xu, Haitao Mi, Yang Liu, and Qun Liu. 2009. Sub-sentence division for tree-based machine translation. In *Proc. od ACL'09 (Short papers)*, pages 137–140.

Appendix A: Crowdsourcing Evaluation Protocol

We follow the protocol proposed by Graham et al. (2016) for evaluation via Amazon Mechanical Turk[8] and use their pre-processing and post-processing software[9]. Adequacy (where the output is compared to the reference) and fluency (where only the output appears) are evaluated independently, according to a 100-point slider, in different experiments. Each of the experiments is composed of 10 HITs (Human Intelligence Tasks) where each HIT includes 100 French sentences which are compared to the reference in the case of adequacy and separately evaluated in the case of fluency. These 100 sentences include 70 MT system outputs extracted randomly from the test set, 10 reference translations, corresponding to 10 of the 70 system outputs, 10 bad reference translations, corresponding to a different 10 of the 70 system outputs and 10 repeat MT system outputs, drawn from the remaining 50 of the original 70 system outputs. The role of the references, bad references and repeat outputs is to control the quality of the evaluation and to not consider ratings from annotators who don't pass the threshold, based on these two main assumptions (see (Graham et al., 2016) for more details):
A: When a consistent judge is presented with a set of assessments for translations from two systems, one of which is known to produce better translations than the other, the score sample of the better system will be significantly greater than that of the inferior system.
B: When a consistent judge is presented with a set of repeat assessments, the score sample across the initial presentations will not be significantly different from the score sample across the second presentations. We here require that each HIT will be answered by 10 different annotators, who are self-assessed native French speakers.

Main setting (§4 and §5): In each of the two crowdsourcing experiments, which correspond respectively to the evaluation of adequacy and fluency in the case where only Rule #1 is applied, we include 10 systems: 4 Transformer systems, namely Transformer SemSplit in both FullTrain and LessTrain settings and the corresponding baselines; 4 LSTM systems (LSTM SemSplit in the two settings and the corresponding baselines) and 2 phrase-based systems (Moses in the LessTrain setting and its corresponding baseline).

[8] https://www.mturk.com/

[9] https://github.com/ygraham/crowd-alone

Low resource setting (§7): In each of the adequacy and fluency experiments, 12 systems are involved: the Transformer SemSplit, LSTM SemSplit and Moses SemSplit systems and their corresponding baselines, each implemented in both 1MTrain and 100KTrain settings.

Splitting exploration setting (§7): In each of the adequacy and fluency experiments, we include 12 systems: 6 Transformer systems, namely Transformer SemSplit$_{1+2}$ in both FullTrain and LessTrain settings, the corresponding baselines, as well as the two neural splitting systems, 4 LSTM systems (LSTM SemSplit$_{1+2}$ in the two settings and the corresponding baselines) and 2 phrase-based systems (Moses in the LessTrain setting and its corresponding baseline).

Appendix B: Manual Analysis Protocol

We perform a manual result analysis of an extract of the data, using the following protocol. First, we sub-sample 150 sentences from the original test set (3003 sentences), such that it includes the same proportion of sentences that contain 0 to 10 words (18% of sentences), 10 to 20 words (35% of the sentences), 20 to 30 words (28%), 30 to 40 words (14%), 40 to 50 words (4%) and more than 50 words (1%), as in the original corpus. Then, to abstract away from possible parsing errors, the new corpus is manually annotated by a single expert UCCA annotator using the UCCAApp annotation tool (Abend et al., 2017). For each of the 150 sentences, each scene segmentation (according to the UCCA manual annotation) is compared to the Transformer SemSplit output and the Transformer Baseline output for this sentence by a another annotator with high proficiency in both English and French (one of the authors of the paper) to analyze the relative preservation of the input scenes in the two systems. We use a 3 point Likert scale for the comparison, assessing if the SemSplit scene preservation is worse, similar or better, compared to the baseline. In the same way, the cohesion of the outputs (defined as the links between their different parts) is also compared using a 3 point Likert scale.

Appendix C: Rule #2 in Direct Semantic Splitting (Sulem et al., 2018b)

Minimal Centers in UCCA (Abend and Rappoport, 2013): With respect to units which are not scenes, the category Center denotes the semantic head. For example, "dogs" is the center of the expression "big brown dogs", and "box" is the center of "in the box". There could be more than one Center in a unit, for example in the case of coordination, where all conjuncts are Centers. Sulem et al. (2018b) defined the minimal center of a UCCA unit u to be the UCCA graph's leaf reached by starting from u and iteratively selecting the child tagged as Center.

Rule #2: Given a sentence S, the second rule extracts Elaborator scenes and corresponding minimal centers. The Elaborator scenes are then concatenated to the original sentence where the embedded scenes, except for the minimal center they elaborate are removed. Pronouns such as "who", "which" and "that" are also removed.

Formally, if $\{(Sc_1, C_1) \cdots (Sc_n, C_n)\}$ are the Elaborator scenes of S and their corresponding minimal centers, the rewrite is

$$S \longrightarrow S - \bigcup_{i=1}^{n}(Sc_i - C_i)|Sc_1| \cdots |Sc_n$$

where $S - A$ is S without the unit A. For example, in the case of Elaborator scenes, this rule converts the sentence "He observed the planet which has 14 known satellites" to "He observed the planet| Planet has 14 known satellites.".

After the extraction of Parallel scenes and Elaborator scenes, the resulting simplified Parallel scenes are placed before the Elaborator scenes.

Appendix D: Low-resource Settings

System		Adequacy	Fluency
Transformer$_{1M}$	**Baseline**	56.4	69.6
	SemSplit	47.1	64.0
LSTM$_{1M}$	**Baseline**	52.3	66.5
	SemSplit	46.0	65.5
Moses$_{1M}$	**Baseline**	45.3	65.6
	SemSplit	38.6	61.2
Transformer$_{100K}$	**Baseline**	29.9	56.5
	SemSplit	29.8	57.6
LSTM$_{100K}$	**Baseline**	37.1	56.7
	SemSplit	29.9	60.0
Moses$_{100K}$	**Baseline**	39.3	60.4
	SemSplit	34.5	61.2

Table 4: Raw system adequacy and fluency scores of the SemSplit systems and the baselines on the 1MTrain (top) and 100KTrain (bottom) settings.