Context-Aware Automatic Text Simplification

Kim Cheng SHEANG
Large Scale Text Understanding Systems Lab, TALN / DTIC
Universitat Pompeu Fabra, Barcelona, Spain
kimcheng.sheang@upf.edu

Abstract: Text simplification is the process of transforming complex text into simple text while retaining its original meaning. Simplified text is easy to read and understand by different groups of people, especially children, non-native speakers, and people with mental disabilities (autism, aphasia, dyslexia). The biggest challenge in Text Simplification is to get the text as simple as possible while preserving the same meaning. In this paper, we are proposing a new research on Text Simplification, which follows human behaviour by taking into account the context from surrounding sentences.
Keywords: Text Simplification, Complex Word Identification, Natural Language Processing

1 Introduction

Text simplification (TS) aims to reduce the complexity of the text while preserving its original meaning. Research on TS has gained its momentum in the last few decades because of its benefits as a tool for reading aids, which could make the information more accessible to broader audiences (Saggion, 2017), or help improve the performance of other Natural Language Processing (NLP) tasks. TS has been shown useful for developing reading aids for children (Watanabe et al., 2009; Siddharthan, 2002), non-native speakers (Siddharthan, 2002), people with cognitive disabilities such as autism (Barbu et al., 2015; Orăsan, Evans, and Dornescu, 2013), aphasia (Carroll et al., 1999) or dyslexia (Rello et al., 2013; Matausch and Peböck, 2010). Moreover, TS can also be used as a preprocessing step to improve the results of many NLP tasks, e.g., Parsing (Chandrasekar, Doran, and Srinivas, 1996), Information Extraction (Evans, 2011; Jonnalagadda and Gonzalez, 2010), Question Generation (Bernhard et al., 2012), Text Summarization (Siddharthan, Nenkova, and McKeown, 2007), and Machine Translation (Štajner and Popović, 2016).

Generally, there are two components of TS: Lexical Simplification (LS) (word-level simplification) and Syntactic Simplification (sentence-level simplification).

LS simplifies text mainly by substituting difficult and less frequently-used words with simpler equivalents. Usually, the pipeline of LS comprises the following steps: complex word identification, substitution generation, substitution selection, and substitution ranking (Paetzold, Specia, and Bank, 2016). More often, LS is regarded as the simplest of all TS sub-tasks; however, it is still a very challenging task because the substitution needs to make sure that both the meaning and grammaticality are well preserved.

Sentence simplification involves transforming both lexical and syntactic structure of the sentences, which help improve readability and comprehension. The most common operations of syntactic simplification are splitting, reordering, dropping, and substituting. These processes often need hand-crafted rules, which are hard to define; moreover, these rules are language-dependent, so porting the system to another language requires rewriting new rules. In recent years, new approaches using Machine Translation (MT), which do not require hand-crafted rules, have become quite popular and achieved good results such as Neural MT (NMT) (Nisioi et al., 2017; Wang et al., 2016), syntax-based MT (SBMT) (Xu et al., 2016), phrase-based MT (PBMT) (Štajner, Calixto, and Saggion, 2015; Coster and Kauchak, 2011; Wubben, van den Bosch, and Krahmer, 2012; Specia,

Lloret, E.; Saquete, E.; Martínez-Barco, P.; Sepúlveda-Torres, R. (eds.) *Proceedings of the Doctoral Symposium of the XXXV International Conference of the Spanish Society for Natural Language Processing (SEPLN 2019)*, p. 56–62
Bilbao, Spain, September 25th 2019.

2010), and tree-based MT (TBMT) (Zhu, Bernhard, and Gurevych, 2010; Woodsend and Lapata, 2011).

In this paper, we describe our proposed TS and some experiments related to Complex Word Identification we have done thus far.

In section 2, we give an overview of TS. Section 3, we describe all the details about our experiments on CWI. Section 4, 5, and 6, we talk about our proposed methodology for TS along with some challenges and future work.

2 Related Work

Rule-based TS was first proposed by Chandrasekar, Doran, and Srinivas (1996) and later by Siddharthan (2002). Sentence simplification using rule-based requires a lot of handcrafted rules for sentence splitting such as the rules to separate relative clauses, coordinate clauses, and remove appositives. The limitation of this approach is that it requires a lot of hand-crafted rules, which are language-dependent and tough to define.

Other approaches have explored TS as a monolingual translation problem (Wubben, van den Bosch, and Krahmer, 2012; Coster and Kauchak, 2011; Zhu, Bernhard, and Gurevych, 2010), utilizing corpora like WikiSmall (Zhu, Bernhard, and Gurevych, 2010), Simple English Wikipedia (SEW), and other paraphrase databases (PPDB). These models are trained on aligned sentences extracted from the corpus, which contains the transformation information needed for the simplification such as reordering, insertion, and deletion. There are many approaches based on statistical Machine Translation (SMT), including phrase-based MT (PBMT) (Štajner, Calixto, and Saggion, 2015; Coster and Kauchak, 2011; Wubben, van den Bosch, and Krahmer, 2012), tree-based MT (TBMT) (Zhu, Bernhard, and Gurevych, 2010; Woodsend and Lapata, 2011), and syntax-based MT (SBMT) (Xu et al., 2016). Phrase-based MT (PBMT) was first introduced by Koehn, Och, and Marcu (2003) then was first used for TS by Specia (2010) and received good result on LS. Syntax-based MT (SBMT) was used by Xu et al. (2016) for sentence simplification, using a large scale PPDB (Ganitkevitch, Durme, and Callison-Burch, 2013), which was extracted from bilingual parallel corpora containing over 100 million sentence pairs and over 2 billion English words.

Compared with SMT, neural MT-based systems (Nisioi et al., 2017; Zhang and Lapata, 2017) have been shown to produce better results. Nisioi et al. (2017) introduced NTS NMT-based system and reported better performance over PBMT in terms of BLUE score and human evaluation. Zhang and Lapata (2017) took a similar approach adding lexical constraints combining the NMT model with reinforcement learning.

Zhao et al. (2018) has recently introduced two new sentence simplification approaches based on neural network. Both approaches are based on a multi-layer and multi-head attention architecture called Transformer (Vaswani et al., 2017) and integrated with the Simple PPDB, an external sentence simplification knowledge base. The results show that the new models outperform all previous state-of-the-art models in sentence simplification.

3 Complex Word Identification

Complex Word Identification (CWI) is the first step in LS (Paetzold and Specia, 2015), used for identifying the difficult words that should be simplified.

Camb (Gooding and Kochmar, 2018) is currently the state-of-the-art CWI for monolingual English datasets. The system uses lexical feature such as number of characters, number of syllables, number of synonyms, word n-gram, POS tags, dependency parse relations, number of words grammatically related to the target word, and Google n-gram word frequencies. Also, they used psycholinguistic features such as word familiarity rating, number of phonemes, imageability rating, concreteness rating, number of categories, samples, written frequencies, and age of acquisition. The limitation of this approach is that it is hard to port from one language to another.

TMU (Kajiwara and Komachi, 2018), the current state-of-the-art CWI for Spanish and German, has been developed for multilingual and cross-lingual CWI systems. The systems are implemented using word frequencies features extracted from the learner corpus (Lang-8 corpus) Mizumoto et al. (2011), Wikipedia and WikiNews. The features contain the number of characters, the number of words, and the frequency of the target word.

NLP-CIC (Aroyehun et al., 2018) developed the systems for both English and Spanish

using binary classification and deep learning (Convolution Neural Network). The feature-based approach uses features such as word frequency of the target word from Wikipedia and Simple Wikipedia corpus, syntactic and lexical features, psycholinguistic features and entity features, and word embedding distance as a feature which is computed between the target word and the sentence. The deep learning approach based on CNN uses only word embeddings GloVe (Pennington, Socher, and Manning, 2014) to represent target words and its context. This approach is very simple and achieves the best results over other deep learning approaches.

As a first approach to TS, we have developed a CWI method inspired by Aroyehun et al. (2018) deep learning approach. In the following sections, and before stating how we intend to investigate TS, we present the experiments we have carried out thus far.

3.1 Method

Preprocessing We generate the left context (LC) and the right context (RC) from those words that appear on the left and the right of the target words. We then extract the 300-dimensional vector from pre-trained word embeddings GloVe (Pennington, Socher, and Manning, 2014). For the vector of the LC or RC, we use a 300-dimensional vector calculated as the average of word vectors of the LC or RC extracted from GloVe word embeddings. If the target is located at the beginning or the end of the sentence, we fill the vector with zeros. Next, we generate a matrix where the first row corresponds to the LC vector, the second row corresponds to the RC vector, and the last n rows are the word embedding vectors of the target words. In order to have a fully consistence matrix representation, we pad it with zero vectors. Next, we extract morphological features (word frequency, tf-idf, number of characters and syllables), linguistic features (part-of-speech, dependency), and normalize it between 0-1 then append it to the next column of the matrix (one column per feature).

Architecture and Training Details We train our model using Convolutional Neural Network (CNN) with the number of filters 128, stride of 1, and kernel size of 3,4,5. We apply the ReLu activation function with Max Pooling to the output of this layer (the output is called feature maps). The feature maps are flattened and pass through three Fully-Connected layers (FC). The first two FC layers use ReLu activation function with 256 and 64 of outputs. The last FC layer uses Softmax activation function which provides the output as complex (1) or non-complex (0). For the training, we train the network using Adam optimizer with 0.001 learning rate, batch size of 128, and 200 epochs. For every 20 iterations, we evaluate the model with our development set and save the model if it achieves the highest f1-score.

3.2 Datasets

We use the CWIG3G2 datasets from (Yimam et al., 2017a; Yimam et al., 2017b) for our CWI system for both training and evaluation. The datasets are collected for multiple languages (English, Spanish, German). The English dataset contains news from three different genres: professionally written news, WikiNews (news written by amateurs), and Wikipedia articles. For Spanish and German, they are collected from Spanish and German Wikipedia articles. For English, each sentence is annotated by 10 native and 10 non-native speakers. For Spanish, it is mostly annotated by native speakers, whereas German it is annotated by more non-native than native speakers. Each sentence contains a target text which is selected by annotators, and it is marked as complex if at least one annotator annotates as complex.

Source	Examples		
	Train	Dev	Test
News	14,002	1,764	2,095
WikiNews	7,746	870	1,287
Wikipedia	5,551	694	870
Spanish	13,750	1,622	2,233
German	6,946	795	959

Table 1: CWI datasets

3.3 Evaluation

The evaluation has shown that our model achieves quite similar results as the state-of-the-art system for English and better than the state-of-the-art system for both Spanish and German. Also, our model performs better when it is trained on a larger dataset. For example, the model achieves a score of 86.79 on the English News dataset with 14,002 examples compared to the score of 83.86 on the English WikiNews dataset with 7,746

examples and the score of 80.11 on the English Wikipedia dataset with 5,551 examples.

System	Macro-F1		
	News	WikiNews	Wikipedia
Camb	**87.36**	**84.00**	**81.15**
TMU	86.32	78.73	76.19
NLP-CIC	85.51	82.40	77.20
Our CWI	86.79	83.86	80.11

Table 2: The evaluation results for English

System	Macro-F1	
	Spanish	German
TMU	76.99	74.51
NLP-CIC	76.72	-
Our CWI	**79.70**	**75.89**

Table 3: The results for Spanish and German

4 *Proposed Methodology for TS*

We are proposing a sentence simplification model that utilizes a multi-layer and multi-head attention architecture inspired by Zhao et al. (2018). The model is based on Transformer architecture (Vaswani et al., 2017), as shown in figure 1. Given a normal sentence A with its context sentences and a simple sentence B with its context sentences, the model learns the mapping from A to B. The model will be trained using a collection of Wikipedia datasets from PWPK (Zhu, Bernhard, and Gurevych, 2010), Woodsend and Lapata (2011), and Kauchak (2013). For testing, we will use Newsela (a professional dataset explicitly created for TS) (Xu et al., 2016).

Additionally, in order to help the system identify words that should be simplified, we are planning to use Paraphrase Database for Simplification (PPDB) (Pavlick and Callison-Burch, 2016), as shown in Table 4.

medical practitioner	:	doctor
legislative texts	:	laws
hypertension	:	high blood pressure
prevalent	:	very common
significant quantify	:	a lot
impact negatively	:	be bad

Table 4: Examples of PPDB

5 *Challenges*

Automatic TS is a challenging NLP task, which often requires lexical, syntactic, or discourse level simplification. Moreover, It is not

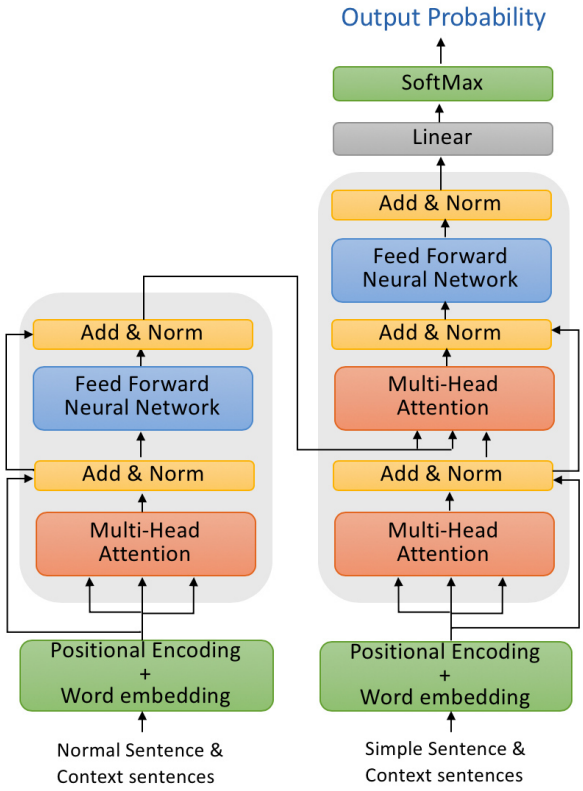

Figure 1: Diagram of the Transformer

easy to decide whether the output generated by a model is good or bad because we do not have any reliable metric to measure the accuracy of the TS system. So far, BLEU (Papineni et al., 2002) and SARI (Xu et al., 2016) are the most commonly used. BLEU, which is borrowed from Machine Translation, is opposed by Sulem, Abend, and Rappoport (2018a) as not suitable for the evaluation of TS. SARI is created for Text Simplification, but still, it is not accurate enough. Recently, another metric called SAMSA (Sulem, Abend, and Rappoport, 2018b) has been created to measure structural aspects of TS. Therefore, very often, we need human judgments to evaluate the TS system, which is hard and time-consuming. Also, the last problem is that targeting each group of users requires a different set of criteria and data.

6 *Future Work*

This work is at a very early stage. We started building the complex word identification system as the experiments for lexical simplification. In the future, for CWI, we would like to try deep contextualize word embeddings, BERT (Devlin et al., 2018), instead of GloVe. Our final goal is to build a good Text Sim-

plification system using the Attention-based model that can understand the context of the whole text.

Acknowledgements

I would like to thank my supervisor, Prof. Horacio Saggion, for the guidance, advice, feedback, and suggestions.

References

Aroyehun, S. T., J. Angel, D. A. Pérez Alvarez, and A. Gelbukh. 2018. Complex Word Identification: Convolutional Neural Network vs. Feature Engineering. pages 322–327.

Barbu, E., M. T. Martín-Valdivia, E. Martínez-Cámara, and L. A. Ureña-López. 2015. Language technologies applied to document simplification for helping autistic people. *Expert Systems with Applications*.

Bernhard, D., L. de Viron, V. Moriceau, and X. Tannier. 2012. Question Generation for French : Collating Parsers and Paraphrasing Questions Question Generation for French : Collating Parsers and Paraphrasing Questions. (January 2012).

Carroll, J., G. Minnen, D. Pearce, and Y. Canning. 1999. Simplifying text for language-impaired readers. *In Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, pages 269–270.

Chandrasekar, R., C. Doran, and B. Srinivas. 1996. Motivations and Methods of Text Simplification. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, number 9, pages 1041–1044.

Coster, W. and D. Kauchak. 2011. Simple English Wikipedia: A New Text Simplification Task William. *Research Journal of Agricultural Sciences*, pages 665–669.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Evans, R. J. 2011. Comparing methods for the syntactic simplification of sentences in information extraction. *Literary and Linguistic Computing*, 26(4):371–388.

Ganitkevitch, J., B. V. Durme, and C. Callison-Burch. 2013. Association for Computational Linguistics PPDB: The Paraphrase Database. *Proceedings of NAACL-HLT 2013*, (June):758–764.

Gooding, S. and E. Kochmar. 2018. CAMB at CWI Shared Task 2018: Complex Word Identification with Ensemble-Based Voting. (2006):184–194.

Jonnalagadda, S. and G. Gonzalez. 2010. BioSimplify: an open source sentence simplification engine to improve recall in automatic biomedical information extraction. *AMIA 2010 Symposium Proceedings*, 2010:351–5.

Kajiwara, T. and M. Komachi. 2018. Complex Word Identification Based on Frequency in a Learner Corpus. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 195–199, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kauchak, D. 2013. Improving Text Simplification Language Modeling Using Unsimplified Text Data. *Acl*, pages 1537–1546.

Koehn, P., F. J. Och, and D. Marcu. 2003. Statistical Phrase-Based Translation. *Acta Neuropathologica*, sep.

Matausch, K. and B. Peböck. 2010. Easyweb - A study how people with specific learning difficulties can be supported on using the internet. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6179 LNCS(PART 1):641–648.

Mizumoto, T., M. Komachi, N. Masaaki, J. Ntt, and Y. Matsumoto. 2011. Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners. Technical report.

Nisioi, S., S. Štajner, S. P. Ponzetto, and L. P. Dinu. 2017. Exploring Neural Text Simplification Models. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91.

Orăsan, C., R. Evans, and I. Dornescu. 2013. Text Simplification for People with Autis-

tic Spectrum Disorders. In *Towards Multilingual Europe 2020: A Romanian Perspective*.

Paetzold, G. and L. Specia. 2015. LEXenstein: A Framework for Lexical Simplification. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 85–90, Stroudsburg, PA, USA. Association for Computational Linguistics and The Asian Federation of Natural Language Processing.

Paetzold, G. H., L. Specia, and W. Bank. 2016. Benchmarking Lexical Simplification Systems. *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 3074–3080.

Papineni, K., S. Roukos, T. Ward, and W.-J. Zhu. 2002. BLEU. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, page 311, Morristown, NJ, USA. Association for Computational Linguistics.

Pavlick, E. and C. Callison-Burch. 2016. Simple PPDB: A Paraphrase Database for Simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–148, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pennington, J., R. Socher, and C. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Stroudsburg, PA, USA. Association for Computational Linguistics.

Rello, L., R. Baeza-Yates, S. Bott, and H. Saggion. 2013. Simplify or help?: text simplification strategies for people with dyslexia. *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, page 15.

Saggion, H. 2017. Automatic Text Simplification. *Synthesis Lectures on Human Language Technologies*, 10(1):1–137, apr.

Siddharthan, A. 2002. An architecture for a text simplification system. In *Language Engineering Conference, 2002. Proceedings*, volume 2002, pages 64–71. IEEE Comput. Soc.

Siddharthan, A., A. Nenkova, and K. McKeown. 2007. Syntactic simplification for improving content selection in multi-document summarization.

Specia, L. 2010. Estimating Machine Translation Post-Editing Effort with HTER. *Proceedings of the AMTA-2010 Workshop Bringing MT to the User: MT Research and the Translation Industry*, (November):33–41.

Štajner, S., I. Calixto, and H. Saggion. 2015. Automatic text simplification for Spanish: Comparative evaluation of various simplification strategies. *International Conference Recent Advances in Natural Language Processing, RANLP*, 2015-Janua:618–626.

Štajner, S. and M. Popović. 2016. Can Text Simplification Help Machine Translation? *In Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, 4(2):230–242.

Sulem, E., O. Abend, and A. Rappoport. 2018a. BLEU is Not Suitable for the Evaluation of Text Simplification. pages 738–744, oct.

Sulem, E., O. Abend, and A. Rappoport. 2018b. Semantic Structural Evaluation for Text Simplification. pages 685–696, oct.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention Is All You Need. jun.

Wang, T., P. Chen, K. Amaral, and J. Qiang. 2016. An Experimental Study of LSTM Encoder-Decoder Model for Text Simplification. sep.

Watanabe, W. M., A. C. Junior, V. R. de Uzêda, R. P. d. M. Fortes, T. A. S. Pardo, and S. M. Aluísio. 2009. Facilita: Reading Assistance for Low-literacy Readers Willian. *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility (W4A) - W4A '10*, page 1.

Woodsend, K. and M. Lapata. 2011. Learning to Simplify Sentences with Quasi-Synchronous Grammar and Integer Programming. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420.

Wubben, S., A. van den Bosch, and E. Krahmer. 2012. Sentence Simplification by Monolingual Machine Translation. *The 50th Annual Meeting of the Association for Computational Linguistics*, 1(January 2014):1015–1024.

Xu, W., C. Napoles, E. Pavlick, Q. Chen, and C. Callison-Burch. 2016. Optimizing Statistical Machine Translation for Text Simplification. *Transactions of the Association for Computational Linguistics*, 4(2011):401–415, dec.

Yimam, S. M., S. Stajner, M. Riedl, and C. Biemann. 2017a. CWIG3G2 – Complex Word Identification Task across Three Text Genres and Two User Groups. pages 401–407.

Yimam, S. M., S. Štajner, M. Riedl, and C. Biemann. 2017b. Multilingual and Cross-Lingual ComplexWord Identification. In *RANLP 2017 - Recent Advances in Natural Language Processing Meet Deep Learning*, pages 813–822. Incoma Ltd. Shoumen, Bulgaria, nov.

Zhang, X. and M. Lapata. 2017. Sentence Simplification with Deep Reinforcement Learning. pages 584–594.

Zhao, S., R. Meng, D. He, S. Andi, and P. Bambang. 2018. Integrating Transformer and Paraphrase Rules for Sentence Simplification. pages 3164–3173.

Zhu, Z., D. Bernhard, and I. Gurevych. 2010. A Monolingual Tree-based Translation Model for Sentence Simplification. *Proceedings of The 23rd International Conference on Computational Linguistics*, (August):1353–1361.