Multilingual Complex Word Identification: Convolutional Neural Networks with Morphological and Linguistic Features

Kim Cheng Sheang
LaSTUS / TALN / DTIC
Universitat Pompeu Fabra
Barcelona, Spain
kimcheng.sheang@upf.edu

Abstract

Complex Word Identification (CWI) is an essential task in helping Lexical Simplification (LS) identify the difficult words that should be simplified. In this paper, we present an approach to CWI based on Convolutional Neural Networks (CNN) trained on pre-trained word embeddings with morphological and linguistic features. Generally, the majority of works on CWI are either feature-engineered or neural network with word embeddings. Both approaches have advantages and limitations, so here we combine both approaches in order to achieve higher performance and still support multilingualism. Our evaluation has shown that our system achieves quite similar performance as the state-of-the-art system for English, and it outperforms the state-of-the-art systems for both Spanish and German.

1 Introduction

Text Simplification (TS) (Saggion, 2017) is a research field which aims at developing solutions to transform texts into simpler paraphrases. Generally, there are two types of TS: Lexical Simplification (lexical-level simplification) and Syntactic Simplification (sentence-level simplification).

The research on TS has become more attractive in recent years because of its benefits as a tool for reading aids or help improve the performance of other Natural Language Processing (NLP) tasks. TS has been shown useful for developing reading aids for children (Siddharthan, 2002; Watanabe et al., 2009), non-native speakers (Siddharthan, 2002), people with intellectual disabilities (Bott et al., 2012; Saggion et al., 2015). Moreover, TS can also be used as a preprocessing step to improve results of many NLP tasks, e.g., Parsing (Chandrasekar et al., 1996), Information Extraction (Evans, 2011; Jonnalagadda and Gonzalez, 2010), Question Generation (Bernhard et al., 2012), Text Summarization (Siddharthan et al., 2004), and Machine Translation (Štajner and Popovic, 2016).

Lexical Simplification (LS) simplifies text mainly by substituting difficult and less frequently-used words with simpler equivalents. Typically, the pipeline of LS comprises the following steps: complex word identification, substitution generation, substitution selection, and substitution ranking (Paetzold and Specia, 2015).

In this work we concentrate on Complex Word Identification (CWI), a core component of LS, which is used to identify difficult words or phrases that are needed to be simplified. Language difficulty often comes at the lexical level, so simply applying the LS alone could help improve reader understanding and information retention (Leroy et al., 2013).

In this paper, we describe our work on CWI based on deep learning approach called Convolutional Neural Networks (CNN) in combination with word embeddings and engineered-features. The task is to create a model that learns from examples and then uses it to classify any target text in a given sentence as complex or non-complex. As it will be shown, our approach achieves state of the art performance in Spanish and German data, and almost state of the art performance in English data.

We carry out our experiments on data from the Complex Word Identification Shared Task 2018 (Yimam et al., 2017b). Here are two examples from the English and Spanish datasets:

> **En**: Both China and the Philippines **flexed their muscles** on Wednesday.

Proceedings of the Student Research Workshop associated with RANLP-2019, pages 83–89,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/issn.2603-2821.2019_013

Es: Allston es un **vecindario** (municipio) de Boston, en los Estados Unidos, ubicado en la parte occidental de la ciudad.

The target text **flexed their muscles** in the English sentence and **vecindario** in the Spanish sentence are annotated as complex by at least one annotator.

In Section 2, we give an overview of recent research on CWI. Section 3, we describe all the details about the implementation of our system. Section 4 is about the details of the datasets we use in the experiments. Section 5, we present the performance of our system with some discussion. Finally, Section 6 is our conclusion and future work.

2 Related Work

There are many different techniques have been introduced so far to identify complex words (Paetzold and Specia, 2016b; Yimam et al., 2018). It is obvious that feature-based approaches remain the best, but deep learning approaches have become more popular and achieved impressive results.

Gooding and Kochmar (2018) proposed a feature-based approach for monolingual English datasets. The system used lexical features 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. It also 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 model achieved the state-of-the-art results for English datasets during the CWI Shared Task 2018 (Yimam et al., 2018), but the limitation of this approach is that it is hard to port from one language to another.

Kajiwara and Komachi (2018) developed a system for multilingual and cross-lingual CWI. The system was implemented using word frequencies features extracted from the learner corpus (Lang-8 corpus) Mizumoto et al. (2011), Wikipedia and WikiNews. The features contained the number of characters, the number of words, and the frequency of the target word. The system achieved state-of-the-art results for both Spanish and German datasets.

Aroyehun et al. (2018) developed systems for both English and Spanish using binary classification and deep learning (CNN) approaches. The feature-based approach used 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 used GloVe word embeddings (Pennington et al., 2014) to represent target words and its context. The deep learning approach is very simple and achieves better results than other deep learning approaches.

Our methodology follows that of Aroyehun et al. (2018) deep learning model in combination with word embeddings and linguistic features.

3 Model

In this section, we explain our approach based on Convolutional Neural Networks (CNN) trained on word embeddings and engineered features. Section 3.2 describes the details on how to preprocess data, transforming from a raw sentence into a matrix of numbers containing all the features described in Section 3.1. Section 3.3 describes the overall architecture of our network, Hyperparameters tuning and training details.

3.1 Features

In this section, we describe all features incorporated in our system.

Word Embeddings Feature: We use pretrained word embeddings GloVe (Pennington et al., 2014) with 300 dimensions to extract word vector representation of each word for all the three languages. For English, we use the model trained on Wikipedia 2014 and Gigaword 5 model (6B tokens, 400K vocab).[1] For Spanish, we use the model (Cardellino, 2016) trained on 1.5 billion words data from different sources: dumps from the Spanish Wikipedia, Wikisource, and Wikibooks on date 2015-09-01, Spanish portion of SenSem, Spanish portion of Ancora Corpus, Tibidabo Treebank and IULA Spanish LSP Treebank, Spanish portion of the OPUS project corpora, and Spanish portion of the Europarl.[2] For German, we use

[1] https://nlp.stanford.edu/projects/glove

[2] https://github.com/dccuchile/spanish-word-embeddings

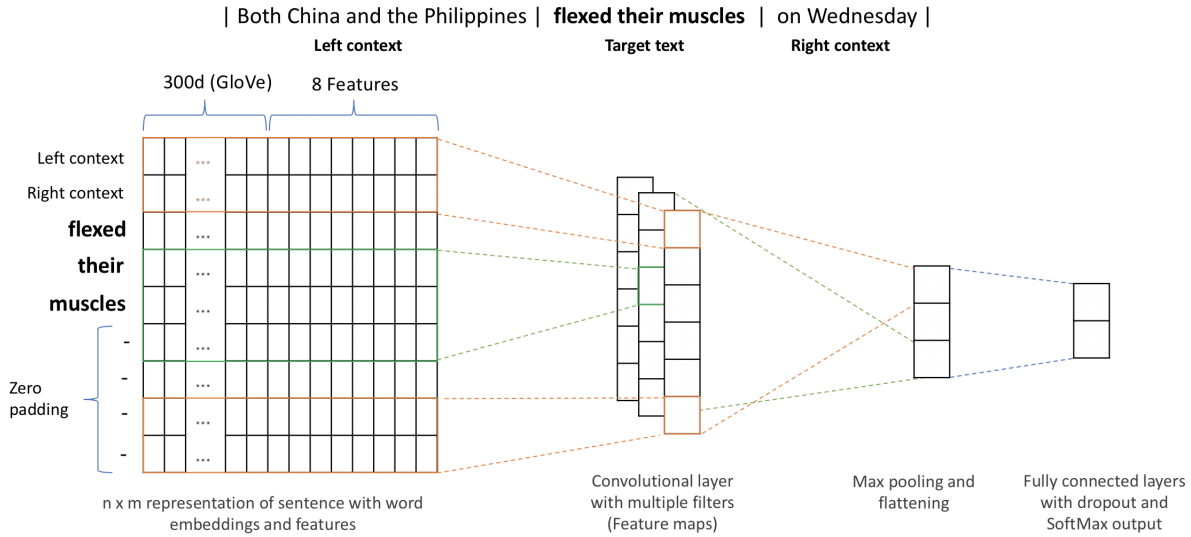

Figure 1: The model architecture

the model trained on the latest dumps of German Wikipedia.[3]

Morphological Features: Our morphological feature set consists of word frequency, word length, number of syllables, number of vowels, and tf-idf.

- Word frequency: the frequency of each word is extracted from the latest Wikipedia dumps as the raw count and then normalize to between 0 and 1.
- Word length: the number of character in the word.
- Number of syllables: the number of syllables of the word, calculated using Pyphen.[4]
- Number of vowels: the number of vowels in the word.
- tf-idf: Term frequency - inverse document frequency, calculated using scikit-learn library.[5]

Linguistic Features: The linguistic features consists of part-of-speech, dependency, and stop word.

- Part-of-speech (POS): a category to which a word is assigned in accordance with its syntactic functions, e.g. noun, pronoun, adjective, verb, etc.
- Dependency: a syntactic structure consists of relations between words, e.g. subject, preposition, verb, noun, adjective, etc.
- Stop word: a commonly used word such as "the", "a", "an", "in", "how", "what", "is", "you", etc.

All these features are extracted using SpaCy (Honnibal and Montani, 2017).

3.2 Preprocessing

We separate each sentence into three parts: target text, left context and right context. The target text is a word or a phrase which is selected and marked as complex or non-complex by the annotators. The left context and the right context are words that appear to the left and the right of the target text.

First, we remove all special characters, digits, and punctuation marks. Then, each word is replaced by its word vector representation using pre-trained word embeddings from the GloVe model as described in Section 3.1. Words that do not exist in the pre-trained word embeddings are replaced with zero vector. Afterward, we transform left context and right context into a 300-dimensional vector calculated as the average of the vectors of all the words in the left context and the right context. If left context or right context is empty (when the target text is at the beginning or the end of the sentence), we replace it with a zero vector. Next, we initialize a matrix X of size $n * m (n = h + 2, m = 308)$ where the first row corresponds to the left context vector, the second

[3] https://deepset.ai/german-word-embeddings
[4] https://pyphen.org
[5] https://scikit-learn.org

row corresponds to the right context vector, and the last r rows are given by the embedding vectors of the words contained in the target text, where r is the number of words in the target text. In order to have a fixed size matrix, we pad the remaining rows p with zero vectors, where $p = h - r$ and h is the maximum value of r in the corpus.

To convert each feature into a vector representation, first we need to transform its values. For example:

- Part-of-speech and Dependency have values such as N, V, ADJ, ADV, and PREP, so we index as 1, 2, 3, 4, 5 and normalize it to between 0 and 1.
- Stop word: 1-stop word, 0-otherwise.
- All the values of word frequency, word length, number of syllables, number of vowels, and tf-idf are numbers, so we just normalize it to between 0 and 1.

For each feature, we initialize a matrix of one column and n rows where the first row corresponds to the average value of the left context, the second row corresponds to the average value of the right context, and the last r rows are the values of the feature for each word in the target text, and the remaining rows are padded with zero. Then, we append this matrix to the previous matrix X.

3.3 Hyperparameters and Training

Figure 1 shows the general architecture of our network. The model has been constructed using pure Tensorflow deep learning library version 1.14.[6]

We train our model using CNN with the number of filters 128, stride of 1, and kernel size of 3, 4, and 5. We then apply the ReLu activation function with Max Pooling to the out of this layer; the output of this layer is often called feature maps. The feature maps are flattened and pass through three Fully-Connected layers (FC) with dropout between each layer. 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 all datasets, the training is done through Stochastic Gradient Descent over shuffle mini-batches using Adam optimizer (Kingma and Ba, 2014) with the learning rate of 0.001, dropout rate of 0.25, mini-batch size of 128. Also, we use weighted cross-entropy as a loss function with the weight of 1.5 for the positive since our datasets are imbalanced; it contains roughly 60% negative examples and 40% positive examples as you can see in the Table 1. We train the system for 200 epochs, and for every 20 iterations, we validate the system with the shuffle development set. Then, if the model achieves the highest f1-score, we save the model and use it for our final evaluation with the test set. In our case, all the hyperparameters are selected via a grid search over the English development set.

We train and evaluate each language separately. For English, the dataset has three different genres, so we combine and train all at once. For Spanish and German, it has only one genre, so we use it directly for training.

[6] https://www.tensorflow.org

4 Datasets

Table 1 shows all the details about each dataset used in the experiments.

Dataset	Train	Dev	Test	Positive
News	14,002	1,764	2,095	40%
WikiNews	7,746	870	1,287	42%
Wikipedia	5,551	694	870	45%
Spanish	13,750	1,622	2,233	40%
German	6,151	795	959	42%

Table 1: English, Spanish and German datasets

We use the CWIG3G2 datasets from (Yimam et al., 2017a,b) 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.

System	English			Spanish	German
	News	WikiNews	Wikipedia		
Camb (Gooding and Kochmar, 2018)	**87.36**	**84**	**81.15**	-	-
TMU (Kajiwara and Komachi, 2018)	86.32	78.73	76.19	76.99	74.51
NLP-CIC (Aroyehun et al., 2018)	85.51	83.08	77.2	76.72	-
ITEC (De Hertog and Tack, 2018)	86.43	81.10	78.15	76.37	-
NILC (Hartmann and Santos, 2018)	86.36	82.77	79.65	-	-
CFILT_IITB (Wani et al., 2018)	84.78	81.61	77.57	-	-
SB@GU (Alfter, 2018)	83.25	80.31	78.32	72.81	69.92
Gillin Inc.	82.43	70.83	66.04	68.04	55.48
hu-berlin (Popović, 2018)	82.63	76.56	74.45	70.80	69.29
UnibucKernel (Butnaru and Ionescu, 2018)	81.78	81.27	79.19	-	-
LaSTUS/TALN (AbuRa'ed and Saggion, 2018)	81.03	74.91	74.02	-	-
Our CWI	86.79	83.86	80.11	**79.70**	**75.89**

Table 2: The evaluation results

5 Results

Table 2 shows the results of our model against others (all the results are based on macro-averaged F1-score).

Our evaluation has shown that when training with the dataset which has more training examples, the model achieves the better result. For example, the model achieves the 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.

We have found an interesting problem. A word can be both complex and non-complex in the same sentence, depending on the selection of the target text. Consider the following sentence, for example,

> The distance, chemical composition, and age of Teide 1 could be established because of its membership in the young Pleiades star cluster.

- The target text "**Pleiades**" is annotated by 3 native and 2 non-native speakers as complex, and our system also predicts it as complex.
- The same sentence with different target text "**Pleiades star cluster**". None of native and non-native speakers annotate it as complex, but our system predicts it as complex.

Here is another example,

> Definitions have been determined such that the 'super casino' will have a minimum customer area of 5000 square metres and at most 1250 unlimited-jackpot slot machines.

- For the target text "**casino**", none of native and non-native speakers annotate it as complex, and our system also predicts it as non-complex.
- The same sentence with different target text "**super casino**". Only one non-native speaker annotates it as complex, so it is marked as complex, but our system predicts it as non-complex.

6 Conclusion and Future Work

In this paper, we have presented a new CWI approach that utilizes deep learning model (CNN) with word embeddings and engineered features. The evaluation has shown that our model performs quite well compared to the state-of-the-art system for English, which realizes on feature-engineered, and better than the state-of-the-art systems for both Spanish and German.

In future work, we plan to use deep contextualized word representations such as BERT (Devlin et al., 2018) or XLNet (Yang et al., 2019) instead of GloVe. Also, we plan to add more features which will be extracted from MRC psycholinguistics database (Paetzold and Specia, 2016a) such as age of acquisition, familarity, concretness, and imagery.

Acknowledgments

I would like to give special thanks to my supervisor, Prof. Horacio Saggion, for the advice, feed-

back, and suggestions. Also, I would like to thank the anonymous reviewers for their constructive comments.

References

Ahmed AbuRa'ed and Horacio Saggion. 2018. LaSTUS/TALN at Complex Word Identification (CWI) 2018 Shared Task. *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications* pages 159–165. https://doi.org/10.18653/v1/w18-0517.

David Alfter. 2018. SB @ GU at the Complex Word Identification 2018 Shared Task. *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications* pages 315–321.

Segun Taofeek Aroyehun, Jason Angel, Daniel Alejandro Pérez Alvarez, and Alexander Gelbukh. 2018. Complex Word Identification: Convolutional Neural Network vs. Feature Engineering. *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications* pages 322–327. https://doi.org/10.18653/v1/w18-0538.

Delphine Bernhard, Louis De Viron, Véronique Moriceau, and Xavier Tannier. 2012. Question generation for french: collating parsers and paraphrasing questions. *Dialogue & Discourse* 3(2):43–74.

Stefan Bott, Luz Rello, Biljana Drndarevic, and Horacio Saggion. 2012. Can Spanish be simpler? LexSiS: Lexical simplification for Spanish. In *Proceedings of COLING 2012*. Mumbai, India, pages 357–374.

Andrei M Butnaru and Radu Tudor Ionescu. 2018. UnibucKernel : A kernel-based learning method for complex word identification. *Proceedings ofthe Thirteenth Workshop on Innovative Use ofNLP for Building Educational Applications* pages 175–183.

Cristian Cardellino. 2016. Spanish Billion Words Corpus and Embeddings. https://crscardellino.github.io/SBWCE/.

R Chandrasekar, Christine Doran, and B Srinivas. 1996. Motivations and Methods of Text Simplification. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*. 9, pages 1041–1044. https://www.aclweb.org/anthology/C96-2183 http://portal.acm.org/citation.cfm?id=993361.

Dirk De Hertog and Anaïs Tack. 2018. Deep Learning Architecture for Complex Word Identification. *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications* pages 328–334. https://doi.org/10.18653/v1/w18-0539.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Richard J. Evans. 2011. Comparing methods for the syntactic simplification of sentences in information extraction. *Literary and Linguistic Computing* 26(4):371–388. https://doi.org/10.1093/llc/fqr034.

Sian Gooding and Ekaterina Kochmar. 2018. Camb at cwi shared task 2018: Complex word identification with ensemble-based voting. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. pages 184–194.

Nathan Hartmann and Leandro Borges Santos. 2018. NILC at CWI 2018: Exploring Feature Engineering and Feature Learning. *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications* pages 335–340. https://doi.org/10.18653/v1/w18-0540.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Siddhartha Jonnalagadda and Graciela Gonzalez. 2010. Sentence simplification aids protein-protein interaction extraction. *arXiv preprint arXiv:1001.4273* .

Tomoyuki Kajiwara and Mamoru Komachi. 2018. Complex Word Identification Based on Frequency in a Learner Corpus. *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications* pages 195–199. https://doi.org/10.18653/v1/W18-0521.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv e-prints* page 103. https://doi.org/10.1145/1830483.1830503.

Gondy Leroy, James E Endicott, David Kauchak, Obay Mouradi, and Melissa Just. 2013. User evaluation of the effects of a text simplification algorithm using term familiarity on perception, understanding, learning, and information retention. *Journal of medical Internet research* 15(7):e144. https://doi.org/10.2196/jmir.2569.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining revision log of language learning sns for automated japanese error correction of second language learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*. pages 147–155.

Gustavo Paetzold and Lucia Specia. 2015. LEXenstein: A Framework for Lexical Simplification. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*. Association for Computational Linguistics and The Asian Federation of Natural Language Processing, Stroudsburg, PA, USA, pages 85–90. https://doi.org/10.3115/v1/P15-4015.

Gustavo Paetzold and Lucia Specia. 2016a. Inferring psycholinguistic properties of words. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies. pages 435–440.

Gustavo Paetzold and Lucia Specia. 2016b. Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. pages 560–569.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

Maja Popović. 2018. Complex word identification using character n-grams. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. pages 341–348.

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

Horacio Saggion, Sanja Štajner, Stefan Bott, Simon Mille, Luz Rello, and Biljana Drndarevic. 2015. Making It Simplext. *ACM Transactions on Accessible Computing* 6(4):1–36. https://doi.org/10.1145/2738046.

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

Advaith Siddharthan, Ani Nenkova, and Kathleen McKeown. 2004. Syntactic simplification for improving content selection in multi-document summarization. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, page 896.

Sanja Štajner and Maja Popovic. 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.

Nikhil Wani, Sandeep Mathias, Jayashree Aanand Gajjam, and Pushpak Bhattacharyya. 2018. The Whole is Greater than the Sum of its Parts : Towards the Effectiveness of Voting Ensemble Classifiers for Complex Word Identification. *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications* pages 200–205.

Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodrigues de Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria 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. https://doi.org/10.1145/1805986.1805997.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237* .

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo H Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. *arXiv preprint arXiv:1804.09132* .

Seid Muhie Yimam, Sanja Štajner, Martin Riedl, and Chris Biemann. 2017a. Cwig3g2-complex word identification task across three text genres and two user groups. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. pages 401–407.

Seid Muhie Yimam, Sanja Štajner, Martin Riedl, and Chris Biemann. 2017b. Multilingual and Cross-Lingual ComplexWord Identification. In *RANLP 2017 - Recent Advances in Natural Language Processing Meet Deep Learning*. Incoma Ltd. Shoumen, Bulgaria, pages 813–822. http://www.acl-bg.org/proceedings/2017/RANLP 2017/pdf/RANLP104.pdf.