CASM - Context and Something More in Lexical Simplification

Atharva Kumbhar
kumbhar.atharva@outlook.com

Prathamesh Mulay
prathumulay@gmail.com

Sheetal Sonawane
sssonawane@pict.edu

Dipali Kadam
ddkadam@pict.edu

Abstract

Lexical Simplification is a challenging task that aims to improve the readability of text for non-native people, people with dyslexia, and any linguistic impairments. It consists of 3 components: 1) Complex Word Identification 2) Substitute Generation 3) Substitute Ranking. Current methods use contextual information as a primary source in all three stages of the simplification pipeline. We argue that while context is an important measure, it alone is not sufficient in the process. In the complex word identification step, contextual information is inadequate, moreover, heavy feature engineering is required to use additional linguistic features. This paper presents a novel architecture for complex word identification that uses a pre-trained transformer model's information flow through its hidden layers as a feature representation that implicitly encodes all the features required for identification. We portray how database methods and masked language modeling can be complementary to one another in substitute generation and ranking process that is built on the foundational pillars of Simplicity, Grammatical and Semantic correctness, and context preservation. We show that our proposed model generalizes well and outperforms the current state-of-the-art on well-known datasets.

1 Introduction

The Lexical Simplification task is an important task for the common people who are having difficulties understanding the language. LS systems can be used to increase the accessibility of information Rodrigo Alarcon (2021) and can help various groups such as non-native speakers, people with linguistic impairments, children, etc. The readability of the text is greatly hindered by the presence of complex words. The words in a text document need to be replaced by identical words without changing their meaning. The Lexical Simplification pipeline generally consists of three major tasks:

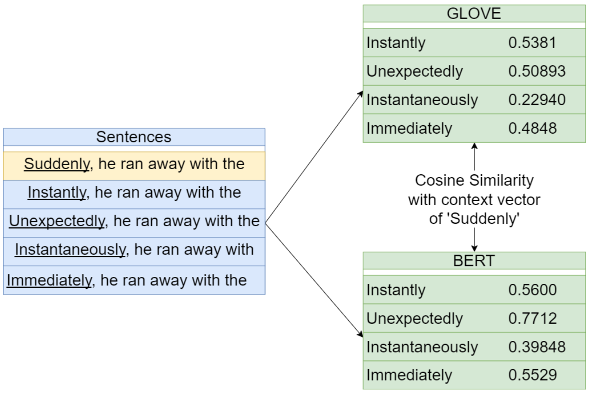
Sentences
Suddenly, he ran away with the
Instantly, he ran away with the
Unexpectedly, he ran away with the
Instantaneously, he ran away with
Immediately, he ran away with the
GLOVE
Instantly 0.5381
Unexpectedly 0.50893
Instantaneously 0.22940
Immediately 0.4848
Cosine Similarity with context vector of 'Suddenly'
BERT
Instantly 0.5600
Unexpectedly 0.7712
Instantaneously 0.39848
Immediately 0.5529

Figure 1: Demonstration of insufficiency in context-dependent approach.

- Complex Word Identification
- Substitute Generation
- Substitute Ranking

Lexical Simplification systems have significantly improved over the past decade. In the early days, such approaches were heavily based on rules and hand-crafted features. Recently, this field has undergone a paradigm shift, and contextual information has become the central theme. The complexity of a word heavily depends on the context in which it was used. The substitutes of the complex words are generated by taking top tokens predicted by the transformer-masked-language model. The model's prediction order is also considered in the ranking of the substitutes. In short, all the stages of the pipeline use contextual information in some ways Gooding and Kochmar (2019); Qiang et al. (2020). However, the context is a central feature, it alone is not sufficient in the process of Lexical Simplification. Figure 1 shows an example of the highlighted word 'suddenly', and the context in which the word appears. The candidate substitutes are of similar word form and even the morphology is of similar stature. This is an example where context alone cannot classify the complex scores accurately, because almost all of the morphological and character level features will be almost the same.

We present CASM-LS (Context and something more), a novel Lexical Simplification pipeline, which uses context and other supporting features in the process. We introduce a novel complex word identification architecture that takes into account the information flow through the hidden layers of the pre-trained transformer models as a feature. The information flow implicitly represents the features required for complexity identification. We capture the information flow by generating a feature map for every word in the sentence by concatenating the outputs of hidden layers of pre-trained transformer models. A sequential model passes layer by layer to output the final complexity. In the substitute generation process, we demonstrate the need to use dictionary-based methods along with masked language models to generate contextual and semantically correct substitutes. When a word is replaced with another word, the contextual representation of the neighbouring words changes, the better the substitute the lesser the change, we employ this as a context-preserving step in the ranking phase. Finally, we employ various ranking methods based on contextual, semantic, grammatical preservation, and simplicity.

The main contributions of the paper are:

- We propose a novel state-of-the-art transformer language model and its information flow-based architecture for complex word identification.
- We demonstrate the necessity of the supplementary methods alongside contextual features in all the stages of the Lexical Simplification pipeline.
- We extensively experiment and evaluate this architecture on various transformer models and benchmark datasets.

2 Related Work

Previous work on Text Simplification focused on hand-crafted rules for simplifying the structure of the sentence Carroll et al. (1998); Siddharthan (2004). Research has shown that Lexical Simplification has a significant role in text simplification Bott et al. (2012); Saggion et al. (2015). The past decade of research has shown that LS is a three-tier pipeline comprising CWI (Complex Word Identification) Shardlow (2013), SG (Substitute Generation) Zhou et al. (2019), and SR (Substitute Ranking).

Early work on CWI used various approaches such as Simplify-Everything(simplifying every word in the sentence) Carroll et al. (1998), Threshold-based, Lexicon based Kajiwara et al. (2013), Implicit CWI Bott et al. (2012); Glavaš and Stajner (2015) and Machine Learning based Paetzold and Specia (2016); Yimam et al. (2018); Kuru (2016); S.P et al. (2016); Bingel et al. (2016). In the 2016 SemEval task of CWI Paetzold and Specia (2016), Machine Learning approaches were explored. 400 non-native English speakers were examined which helps to create a corpus for complex words in sentences. Recent work in CWI has shown that context greatly influences the complexity score of the word in the sentence. Gooding and Kochmar (2018) framed CWI as a sequence labeling task, achieving state-of-the-art results on CWI 2018 dataset. Pan et al. (2021) have used pre-trained transformer models, that better encode the context of the sentence. Bani Yaseen et al. (2021); Pan et al. (2021)have used an ensemble of pre-trained transformer models to classify the complexity of words between 1-5, one being very easy and five being very difficult. They have trained and tested their model on the LCP dataset[1] Desai et al. (2021) which contains data from Bible, Europarl and biomedical literature.

The Substitute Generation process was entirely focused on database-driven methods. The substitutes of the target complex words were found by querying a database and retrieving synonyms. Various databases such as WordNet[2], PPDB Ganitkevitch et al. (2013), DataMuse[3], Big Huge Thesaurus[4], etc have been used in the past. Yatskar et al. (2010) used parallel and aligned corpora of English Wikipedia and Simple English Wikipedia. Glavaš and Stajner (2015) uses word embeddings to get the most semantically similar words.

The challenge of the ranking procedure is to accurately find which of the possible substitutes fit the sentence whilst maintaining grammatical integrity and preserving the meaning of the sentence. De Belder and Moens (2012) use a latent variable language model to implicitly assign senses

[1] https://github.com/MMU-TDMLab/CompLex

[2] https://wordnet.princeton.edu/

[3] https://www.datamuse.com/api/

[4] https://words.bighugelabs.com/site/api

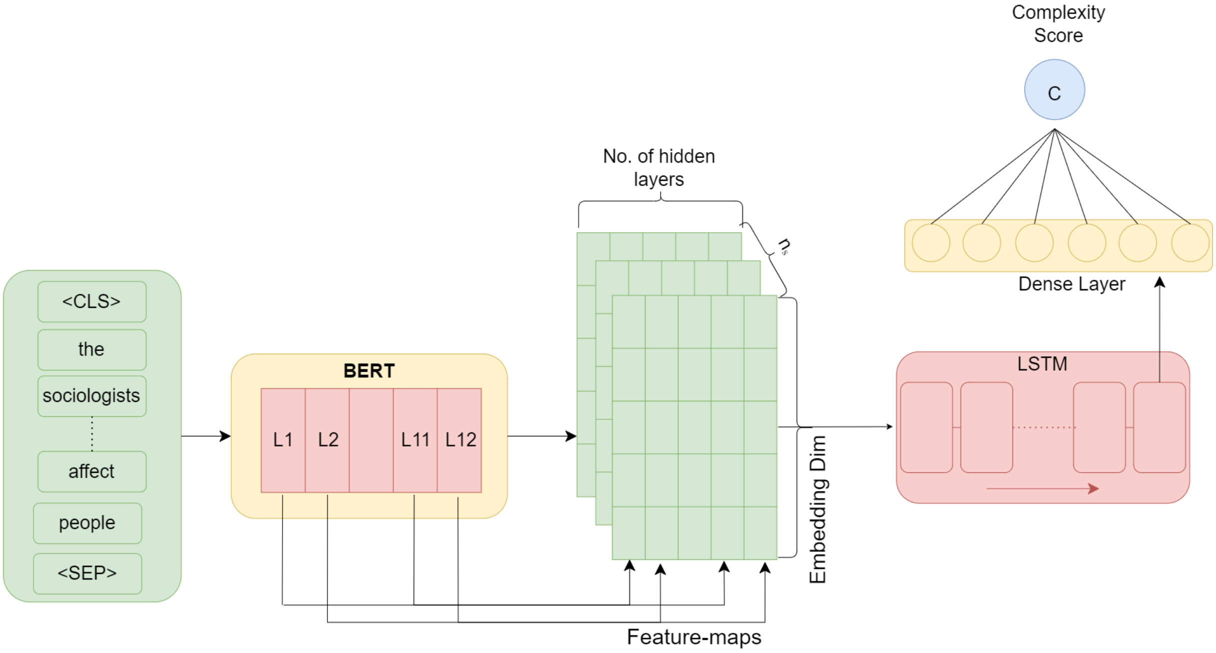

Figure 2: Complex Word Identification Architecture

to the substitutes and discard the senses that do not match with the complex word. Word Sense Disambiguation techniques have been used in the past Anderson (2001), to discard words with a different sense. Most of the techniques use a hybrid approach, by considering syntactic, simplicity, and semantic measures Gooding and Kochmar (2019). Qiang et al. (2020); Wang et al. (2016) uses contextual information at every stage of the Lexical Simplification process. They use the BERT masked language modeling for finding the substitutes of target words and use other BERT features such as its ranking order, and cross-entropy loss for ranking and have achieved outstanding results.

Pre-trained language encoders based on the transformers play a central role in our proposed method. The transformer is a combination of encoder and decoder which uses attention to preserve the context of a word in a sentence. The BERT(Bidirectional Encoder Representations from Transformers) Devlin et al. (2018) is a language encoder that has been specifically trained for next-sentence prediction and masked language modeling. BERT has undergone modifications over time and has variations including RoBERTa Liu et al. (2019), ALBERT Lan et al. (2019), XL-NET Yang et al. (2019), etc.

3 Methodology

In this section, we outline each step of our Lexical Simplification pipeline, namely CWI(complex word identification), SG(Substitute Generation), and SR(Substitute Ranking).

3.1 Complex Word Identification

CWI is a crucial step in Lexical Simplification. Mistaking simpler words as complex can adversely affect the semantics of the sentence. Two approaches have been extensively studied in the past: 1) Machine learning-based methods on extracted word-level features Paetzold and Specia (2016); Yimam et al. (2018); Gooding and Kochmar (2018), and 2) CWI as a sequence labelling task Gooding and Kochmar (2019). Machine learning-based methods require extensive feature engineering Gooding and Kochmar (2018), extracted 6-word level features before training on the AdaBoost and Random forest model. Moreover, these methods suffer from domain shifts when used in practice. CWI framed as a sequence labelling task considers the contextual information of the sentence to make a prediction. We argue that context is an essential feature of perceived complexity but is insufficient to justify the word's complexity.

Noting the shortcomings of the two methods, we infer that we need a representation that implicitly encodes contextual information and other important features without extensive feature engineering. We propose CASM-LS, based on the pre-trained model and the information flow through its hidden layers. The intuition is that there is an inherent difference between the information flow of simple words and complex words. The design choices work well for this scenario because:

- The BERT model efficiently encodes the contextual information in the word representations.
- The information flow through the hidden layers implicitly encapsulates other features such as word frequency, morphological character level features, etc.
- There has been a lot of research on understanding and interpreting transformer's self-attention heads and information flow, work by Wiegreffe and Pinter (2019); Vashishth et al. (2019); Clark et al. (2019) show that transformers encode various linguistic features.

To the best of our knowledge, this is the first method that uses the information flow of BERT directly into an account for classification purposes.

Let ns be the number of tokens in the sentence, nh be the number of hidden layers, $l_{ij} \in R^{768}$ be the token embedding of the ith token and jth layer. The sentence is first passed to the pre-trained model. Then, for every word in the sentence, l_{ij}, $j \in 1, 2, ...nh$ are concatenated to give a feature map of $(nh, 768)$. This feature map is fed to an LSTM (Long short-term memory) Hochreiter and Schmidhuber (1997) model with a logistic head that predicts the binary complexity of the word. This outputs a value between 0-1, which is taken as the complexity score. Figure 2 shows our proposed method. Algorithm1 shows how the model is used at test time to identify the complexity score. We keep a threshold value which if exceeded, the word is then to be simplified.

The LSTM model contains 300 hidden units, followed by a dense layer of 200 units and a logistic output head. We hypothesize that a large number of hidden units in LSTM are essential for it to capture the representation in a higher dimensional word embedding.

We started out with a BERT-base model while experimenting with the architecture and hence we set the pre-trained transformer-based model to be BERT-base-uncased for explanation purposes. We have showcased the influence of the selection of transformer models on the results of the architecture in Table 6. Initially, we had set nh=12(all layers). The number of layers to be used can be treated as a hyperparameter, we have evaluated its influence in Table 7.

Algorithm 1 Complex Word Identification

```
Require: Sentence $s$
Require: Word Piece Tokenizer
Require: pretrained BERT model
  $c \leftarrow ComplexityThreshold$
  $tokens \leftarrow Tokenizer(s)$
  $segid \leftarrow Segment(tokens)$
  $indtok \leftarrow Index(tokens)$
  $output \leftarrow bert(indtok, segid, )$
  for token in tokens do
      for layer in layers do
          $l \leftarrow getlayer(token, layer)$
          $fmap \leftarrow concat(fmap, l)$
      end for
      $cmp \leftarrow cwimodel(fmap)$
      if $cmp > c$ then
          $complexdict.add(\{token : cmp\})$
      end if
  end for
```

3.2 Substitute Generation

Once the complex word(s) have been identified in the sentence, we must come up with a grammatically and syntactically correct replacement while keeping the meaning of the sentence. Most of the earlier techniques relied on the dictionary to identify a complex word's equivalent. Sometimes while using the dictionary method, substitutes generated are not contextually correct. Rather than finding the word in the dictionary and giving it the proper form according to the phrase, the current state of the art advises adopting the way of employing the BERT masking model Qiang et al. (2020). We have used two methods for finding the synonyms

1. BERT MASKLM Devlin et al. (2018)
2. Big Huge Thesaurus

For using BERT MASK Language model, first, we mask the complex word with the symbol [MASK] but if the sentence is parsed directly to the model then it generates semantically incorrect synonyms whilst being grammatically and syntactically correct. To prevent this, we randomly mask some percent of the words in the original sentence and append it to the masked sentence, this procedure was employed in Qiang et al. (2020). The

Sentence	the **enormicity** of this situation is very enormous.					
BERT MASKLM	magnitude	complexity	impact	scale	severity	[..]
BHT	outrageousness	atrocity	enormousness	grandness	immenseness	[..]
Sentence	He will **abjure** his allegiance to the king.					
BERT MASKLM	swear	lose	pledge	give	abandon	[..]
BHT	[ ]					

Table 1: Examples of Substitute Generation by BERTMASKLM and BHT .

majority of semantically incorrect synonyms are eliminated when utilizing this technique.

Apart from BERT MASKLM, we also employ web-based methods. Because, if solely the BERT approach were used, sometimes it would be unable to provide all suitable synonyms for complex words. (Table 1) shows how the two methods can work in complementary to one another. In the first example, more accurate candidates should have been "evilness", "wickedness", etc. For such examples, dictionary-based approaches work well. After experiments, it is inferred that *Big Huge Thesaurus*[3](BHT) and *Datamuse*[2] produce promising results. *Big Huge Thesaurus*[3] is an API that can be used to find synonyms and antonyms. *Big Huge Thesaurus* is based on the WordNet database at Princeton University Fellbaum (1998), the Carnegie Mellon Pronouncing Dictionary, and other sources. Sometimes, the BHT method may not be able to give synonyms for particular parts of speech of that word that is used in a sentence. In Table 1's second example case, we rely on the BERT MASKLM method as no output is generated by BHT. To avoid a shortage of substitutes we are producing 20 substitutes from BERT MASKLM.

3.3 Substitute Ranking

The candidate substitutes generated from the previous step need to be ranked to find the most suitable candidates. The ranking and filtering step is important because all the generated candidates may not be the best substitution, it's important for the grammatical and semantic meaning of the sentence to not change after the substitution. Moreover, the substitute candidate should be simpler than the target word. Previous methods have used word frequency Desai et al. (2021); Qiang et al. (2020) as a measure to determine the simplicity, the more the frequency the simpler the word. However, this method results in discarding some of the simple words. N-gram frequencies were used to check the grammatical integrity Gooding and Kochmar (2019). Glavaš and Stajner (2015) used cosine similarity between the target and candidate word embeddings as a score. Qiang et al. (2020) uses additional context-based measures based on BERT.

We have employed three techniques for ranking:

1. Lexical Complexity score
2. Semantic equivalence
3. Context Preservation

3.3.1 Lexical Complexity score

To ensure that the candidate substitute is a simpler alternative, we have used our 3.1 CWI model for filtering the candidates that increase the complexity of the sentence. The complex words are replaced with the candidate substitutes one by one, and their complexity score is given by the CWI model. The candidates that result in a lower score than the target word are retained, while others are discarded. This rules out the words that would have made the sentence more complex.

3.3.2 Semantic Equivalence

The GloVe Pennington et al. (2014) is a word representation model trained on a large corpus, that captures the linear semantic structures in the words. The semantic equivalence score is calculated by taking the cosine similarity between the complex word and the candidate substitutions Glavaš and Stajner (2015). The candidates below the threshold are then discarded.

3.3.3 Context Preservation

Qiang et al. (2020) used a cross-entropy loss over a window of neighbouring words to implicitly model the n-gram frequency of the candidate with its context. We employed a new technique for preserving the context. We kept a window of size 5 around the target word on both sides as its fixed context. We replace the complex word with the candidate substitute in the sentence, and we recompute the

Parameter	Score with BHT	Score without BHT
ACC@1	0.5227	0.4957
ACC@1@Top1	0.2159	0.225
ACC@2@Top1	0.338	0.3219
ACC@3@Top1	0.4062	0.3931
MAP@3	0.3221	0.2961
MAP@5	0.2366	0.2191
MAP@10	0.1478	0.1347
Potential@3	0.75	0.6894
Potential@5	0.8011	0.772
Potential@10	0.8494	0.8119

Table 2: Results On test dataset of TSAR2022 shared task are evaluated using this evaluation matrix

Metric	JUST-BLUE	DEEP-BLUE	CASM-LS
Pearson score	0.7886	0.7882	0.7140
MAE score	0.0609	0.0610	0.0690
spearman score	0.7369	0.7425	0.6766
RSQ score	0.6172	0.6210	0.5029

Table 3: Evaluation on LCP dataset

BERT embeddings for the words. For each word in the window, we take its cosine distance with its original embedding vector, then we average it over the entire window. The higher the score the better substitute it will be. This score along with the Semantic equivalence score is used to rank the substitutes.

4 Results

In the sections below we showcase the evaluation of our approach to various benchmark datasets and metrics.

4.1 CWI

We test our novel complex word identification architecture on three primary datasets- CWI 2018 Yimam et al. (2018), LCP SemEval2021 Desai et al. (2021), and CEFR-LS Uchida et al. (2018). CWI 2018 consists of data from three domains namely- Wikipedia, Wikinews, and News. The words in the dataset are annotated as complex or not by 10 native and 10 non-native speakers. A ratio is also provided that indicates the percentage of people who annotated the word as complex. The dataset assigns the binary label as 1 even if a single person found it positive. We found that the words with lower ratios such as 0.1 were easier, and were causing ambiguity in the model training and degrading performance, hence we removed those ambiguous words by keeping P==0 and P>=0.2.

Method	WikiNews	Wiki	News
CAMB	0.8400	0.8115	0.8736
SEQ	0.8505	0.8158	0.8763
CASM-LS	**0.8815**	0.8015	**0.9256**

Table 4: F1 score Evaluation on Wikinews, Wikipedia and News dataset

The results in (Table 4) indicate that our model significantly surpasses the SEQ model Gooding and Kochmar (2019) on two out of 3 datasets. The transformer model used for testing purposes was Roberta-large and extracted 12 hidden layers as a feature map. Furthermore, to test the generalization of our model we test our bert-base, 12 configuration model trained collectively on all three genres of CWI 2018 dataset on CEFR. Even with the base BERT model, our architecture outperforms SEQ, demonstrating high generalization (Table 5).

Metric	SEQ	CASM-LS
F1 score	0.8575	0.8936

Table 5: Evaluation on CEFR dataset

SemEval-2021 Task 1: The Lexical Complexity Prediction dataset spans three genres: 1) Europarl, 2) The Bible, and 3) Biomedical Literature. This dataset resolves the problem of hard labels in CWI2018 by using a continuous scale to annotate complexity. The metrics used to evaluate

	Wikipedia			Wikinews		
	PRE	RE	FI	PRE	RE	FI
roberta-large	0.8268	0.7778	0.8015	0.8466	0.9195	0.8815
roberta-small	0.7868	0.7926	0.7896	0.8514	0.8563	0.8638
bert-base	0.8738	0.7461	0.8049	0.8909	0.8047	0.8776
bert-large	0.8333	0.7308	0.77866	0.8587	0.9294	0.8926
alberta	0.8319	0.6963	0.7580	0.8448	0.8448	0.8448
alberta-large	0.8962	0.7037	0.7883	0.8817	0.8567	0.8688

Table 6: Evaluation of Influence of Transformer models on Wikipedia and Wikinews dataset

the model performance were Pearson's Correlation, Spearman's Rank, Mean Absolute Error (MAE), Mean Squared Error (MSE), and R2. The results of our model are comparable to the state-of-the-art methods. (Table 3) shows that our model's performance is comparable to the current state of the art. The SOTA systems Bani Yaseen et al. (2021); Pan et al. (2021) on this task have used ensemble methods of large language models and have done extensive feature engineering.

4.2 Substitute Generation and Substitute Ranking

The system's performance is analyzed on the test dataset provided by TSAR 2022 shared task Saggion et al. (2022) on Lexical Simplification is shown in Table 2. The test set contains 373 sentences with complex words taken from the CWI 2018 shared task dataset. The corresponding complex words that need to be simplified were also provided. We have tested the dataset on two approaches. Firstly, we tested using substitutes generated solely with the BertMaskLM method. We include the substitutes generated by big huge thesaurus in the second approach. tested on substitutes generated by BHT and BERTMaskLM both methods we had better results because due to the addition of proper substitutes rather than similar words. The BHT method results in an increased number of potential candidate substitutes which leads to an improvement in potential@3,5,10 metrics by a minimum of 0.03 in each metric.

5 Ablation Study

5.1 Influence of Transformer Models

The transformer used to generate the feature map has a huge influence on the performance of the system. (Table 6) shows the trends in the precision, recall, and f-score values on Wikipedia, and Wikinews datasets. The larger versions of the transformers generally outperform their base versions. The increased model parameters and feature vector dimensions from 768 to 1024 increase the expressivity of the model. The Roberta-large model performs the best among other models.

Layers	PRE	RE	FI
6	0.8302	0.7674	0.8058
8	0.8366	0.7442	0.7876
10	0.8589	0.8140	**0.8454**
12	0.8500	0.7907	0.8294
14	0.8202	0.8488	0.8225
16	0.8023	0.8023	0.8114
18	0.8046	0.8140	0.8148
20	0.8103	0.8198	0.8091

Table 7: Evaluation of the influence of layers on Wikipedia dataset

5.2 Influence of layers

To evaluate the effect of the number of layers used to form a feature map, we used the Roberta-large model and trained the LSTM model on the Wikipedia dataset with different numbers of layers in the feature map. We can observe that the results in table 7 improve with the increase in the number of layers up to 10-12 layers, then it begins to saturate and starts decreasing again.

6 Conclusion

In this paper, we have proposed a novel Lexical Simplification algorithm. We leverage the use of information flow in transformers for complex word identification, which to the best of our knowledge is the first method that directly represents the hidden layers as sequential feature representation. The results show that our system outperforms the current state-of-the-art model on CWI 2018 and CEFR-LS datasets. The importance of supporting features in the simplification process is explored and analyzed.

We showcase a procedure in which the contextual information is infused with other simplicity, semantic and syntactic measures in Substitute Generation and Substitute Ranking. In the future, we plan to expand this novel idea of capturing information flow which can be easily applied to other NLP tasks such as Part-of-Speech tagging and Named Entity Recognition. We also plan to investigate our methodology for text simplification.

Limitations

The Lexical Simplification pipeline being a multistage process, with a high compute cost poses the problem of real-time deployment on large text documents. For instance, the text is processed line by line, and hence it's very inefficient for large documents and articles. The future work is to develop a unified model that solves this problem and is computationally efficient. The BERT MASKED LM technique that we employ concatenates the original sentence in front of the masked sentence and randomly masks some percentage of words. The randomness in this causes the model to output varying results. The lack of large datasets for the CWI process limits its use in domain-specific cases.

References

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, 1997. ISSN 0899-7667. doi:10.1162/neco.1997.9.8.1735.

Carroll, J., Minnen, G., Canning, Y., Devlin, S., and Tait, J. Practical simplification of english newspaper text to assist aphasic readers. *Proc. of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, 1998.

Fellbaum, C., editor. *WordNet: an electronic lexical database*. MIT Press, 1998.

Anderson, R. Lexical morphology and verb use in child first language loss: A preliminary case study investigation. *International Journal of Bilingualism - INT J BILING*, 5:377–401, 2001. doi:10.1177/13670069010050040101.

Siddharthan, A. Syntactic simplification and text cohesion. *Research on Language Computation*, 4, 2004. doi:10.1007/s11168-006-9011-1.

Yatskar, M., Pang, B., Danescu-Niculescu-Mizil, C., and Lee, L. For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 365–368. Association for Computational Linguistics, Los Angeles, California, 2010.

Bott, S., Saggion, H., and Mille, S. Text simplification tools for Spanish. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1665–1671. European Language Resources Association (ELRA), Istanbul, Turkey, 2012.

De Belder, J. and Moens, M.-F. A dataset for the evaluation of lexical simplification. pages 426–437. 2012. ISBN 978-3-642-28600-1. doi:10.1007/978-3-642-28601-8_36.

Ganitkevitch, J., Van Durme, B., and Callison-Burch, C. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764. Association for Computational Linguistics, Atlanta, Georgia, 2013.

Kajiwara, T., Matsumoto, H., and Yamamoto, K. Selecting proper lexical paraphrase for children. In *Proceedings of the 25th Conference on Computational Linguistics and Speech Processing (ROCLING 2013)*, pages 59–73. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP), Kaohsiung, Taiwan, 2013.

Shardlow, M. A comparison of techniques to automatically identify complex words. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, pages 103–109. Association for Computational Linguistics, Sofia, Bulgaria, 2013.

Pennington, J., Socher, R., and Manning, C. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics, Doha, Qatar, 2014. doi:10.3115/v1/D14-1162.

Glavaš, G. and Stajner, S. Simplifying lexical simplification: Do we need simplified corpora? 2015. doi:10.3115/v1/P15-2011.

Saggion, H., Štajner, S., Bott, S., Mille, S., Rello, L., and Drndarevic, B. Making it simplext: Implementation and evaluation of a text simplification system for spanish. *ACM Trans. Access. Comput.*, 6(4), 2015. ISSN 1936-7228. doi:10.1145/2738046.

Bingel, J., Schluter, N., and Martínez Alonso, H. CoastalCPH at SemEval-2016 task 11: The importance of designing your neural networks right. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1028–1033. Association for Computational Linguistics, San Diego, California, 2016. doi:10.18653/v1/S16-1160.

Kuru, O. AI-KU at SemEval-2016 task 11: Word embeddings and substring features for complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1042–1046. Association for Computational Linguistics, San Diego, California, 2016. doi: 10.18653/v1/S16-1163.

Paetzold, G. and Specia, L. SemEval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569. Association for Computational Linguistics, San Diego, California, 2016. doi:10.18653/v1/S16-1085.

S.P, S., Kumar M, A., and K P, S. AmritaCEN at SemEval-2016 task 11: Complex word identification using word embedding. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1022–1027. Association for Computational Linguistics, San Diego, California, 2016. doi:10.18653/v1/S16-1159.

Wang, T., Chen, P., Amaral, K. M., and Qiang, J. An experimental study of LSTM encoder-decoder model for text simplification. *CoRR*, abs/1609.03663, 2016.

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

Gooding, S. and Kochmar, E. 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. Association for Computational Linguistics, New Orleans, Louisiana, 2018. doi:10.18653/v1/W18-0520.

Uchida, S., Takada, S., and Arase, Y. CEFR-based lexical simplification dataset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA), Miyazaki, Japan, 2018.

Yimam, S. M., Biemann, C., Malmasi, S., Paetzold, G., Specia, L., Štajner, S., Tack, A., and Zampieri, M. A report on the complex word identification shared task 2018. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 66–78. Association for Computational Linguistics, New Orleans, Louisiana, 2018. doi:10.18653/v1/W18-0507.

Clark, K., Khandelwal, U., Levy, O., and Manning, C. D. What does BERT look at? an analysis of bert's attention. *CoRR*, abs/1906.04341, 2019.

Gooding, S. and Kochmar, E. Recursive context-aware lexical simplification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4853–4863. Association for Computational Linguistics, Hong Kong, China, 2019. doi:10.18653/v1/D19-1491.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942, 2019.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019.

Vashishth, S., Upadhyay, S., Tomar, G. S., and Faruqui, M. Attention interpretability across NLP tasks. *CoRR*, abs/1909.11218, 2019.

Wiegreffe, S. and Pinter, Y. Attention is not not explanation. 2019. doi:10.48550/ARXIV.1908.04626.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J. G., Salakhutdinov, R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237, 2019.

Zhou, W., Ge, T., Xu, K., Wei, F., and Zhou, M. BERT-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373. Association for Computational Linguistics, Florence, Italy, 2019. doi:10.18653/v1/P19-1328.

Qiang, J., Li, Y., Zhu, Y., Yuan, Y., and Wu, X. Lsbert: A simple framework for lexical simplification. *CoRR*, abs/2006.14939, 2020.

Bani Yaseen, T., Ismail, Q., Al-Omari, S., Al-Sobh, E., and Abdullah, M. JUST-BLUE at SemEval-2021 task 1: Predicting lexical complexity using BERT and RoBERTa pre-trained language models. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 661–666. Association for Computational Linguistics, Online, 2021. doi:10.18653/v1/2021.semeval-1.85.

Desai, A., North, K., Zampieri, M., and Homan, C. M. LCP-RIT at semeval-2021 task 1: Exploring linguistic features for lexical complexity prediction. *CoRR*, abs/2105.08780, 2021.

Pan, C., Song, B., Wang, S., and Luo, Z. DeepBlueAI at SemEval-2021 task 1: Lexical complexity prediction with a deep ensemble approach. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 578–584. Association for Computational Linguistics, Online, 2021. doi: 10.18653/v1/2021.semeval-1.72.

Rodrigo Alarcon, L. M., Paloma Martínez. Lexical simplification system to improve web accessibility. *IEEE*, 28(1):58755 – 58767, 2021. doi: 10.1109/ACCESS.2021.3072697.

Saggion, H., Štajner, S., Ferrés, D., Sheang, K. C., Shardlow, M., North, K., and Zampieri, M. Findings of the TSAR-2022 shared task on multilingual lexical simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 271–283. Association for

Computational Linguistics, Abu Dhabi, United Arab Emirates (Virtual), 2022.