RCML at TSAR-2022 Shared Task: Lexical Simplification With Modular Substitution Candidate Ranking

Desislava Aleksandrova and **Olivier Brochu Dufour**
CBC/Radio-Canada
dessy.aleksandrova@radio-canada.ca, olivier.brochu.dufour@radio-canada.ca

Abstract

This paper describes the lexical simplification system RCML submitted to the English language track of the TSAR-2022 Shared Task. The system leverages a pre-trained language model to generate contextually plausible substitution candidates which are then ranked according to their simplicity as well as their grammatical and semantic similarity to the target complex word. Our submissions secure 6th and 7th places out of 33, improving over the SOTA baseline for 27 out of the 51 metrics.

1 Introduction

Lexical Simplification (LS) is a means to facilitate reading comprehension for different target audiences such as second language learners, native speakers with low literacy levels or various kinds of neurodivergent conditions and reading impairments.

1.1 Task description

The task of lexical simplification (LS) consists in reducing the lexical complexity of a sentence by replacing one (or more) difficult words or multiword expressions (MWE) with easier to read and understand vocabulary all the while preserving the original sense.

Normally, LS includes the task of complex word identification (CWI) (Paetzold and Specia, 2016) but in the context of the TSAR-2022 Shared Task (Saggion et al., 2022), the word to be simplified is provided. Given a sentence containing a complex word, a system should then return an ordered list (best predictions first) of substitutes (min 0, max 10) for the complex word in its original context. The ordered list of predicted candidates must not contain ties, repetitions or the complex word itself. Predicted candidates must be good contextual fits (semantically and syntactically) as well as have the same morphological inflection as the complex word in the original sentence. A team is allowed to submit 3 runs per track.

*Despite the fog, other flights are reported to have landed safely leading up to the **collision**.*

GOLD: *crash, impact, accident, collision*
RCML: *accident, crash, tragedy, incident*

Figure 1: A complex word in context with gold annotations and predicted substitution candidates

Our team participated in the English track and made 2 submissions.

1.2 Dataset description

The TSAR-2022 Shared Task has provided participants with a trial and test sets (.tsv) from a new multilingual lexical simplification dataset (Stajner et al., 2022) in English, Spanish and Portuguese. The trial set of each language contains 10 sentences accompanied by the complex word to simplify (in the second column) and the suggested substitution candidates (24 or 25) in the remaining columns. The test set, in contrast, contains only the first two columns (sentence and complex word). The English test set contains 373 instances (rather than the initially stated 386). The gold test set in English contains multiple simplification suggestions provided by annotators (25 or 26 in some cases).

To produce the dataset, crowdsourced workers were presented with instances (sentences) in which a single token (and never a MWE) is marked as requiring simplification. They were asked to provide simpler synonyms for the marked words, taking into account that the original meaning of the sentence should be preserved. Annotators were allowed to return multiple words if they could not think of a relevant single-word simplification. A number of suggestions match the complex word, since annotators were instructed to submit the com-

Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022), pages 259 - 263
December 8, 2022 ©2022 Association for Computational Linguistics

plex word whenever they couldn't find a simpler substitution. However, the evaluation script ignores such suggestions when calculating the scores.

1.3 Evaluation metrics

The evaluation metrics used in the TSAR-2022 Shared Task are the following:

Mean Average Precision @ K: K={1,3,5,10}. MAP@K evaluates the relevance of the predicted substitutes as well as the position of the relevant candidates compared to the gold annotations.

Potential@K: K={1,3,5,10}. Potential@K evaluates the percentage of instances for which at least one of the substitutions predicted is present in the set of gold annotations.

Accuracy@K@top1: K={1,2,3}. ACC@K@top1 evaluates the ratio of instances where at least one of the K top predicted candidates matches the most frequently suggested substitute in the gold list of annotated candidates.

2 System Description

We propose a modular system for lexical simplification which requires no training data and allows to fine-tune each module separately in order to improve the final result. Since the dataset already provides complex word annotations, RCML is composed of only two modules: one for candidate generation and one for candidate ranking.

2.1 Candidate Generation

To generate substitution candidates, we used the lexical substitution framework LexSubGen (Arefyev et al., 2020) and in particular, the best performing estimator XLNet+emb which employs a target word injection method different to LSBert's (Qiang et al., 2020). To produce a list of substitutes with their probabilities from the XLNet-large-cased model, LexSubGen combines a representation of the original input sentence (without masking) with the product of two distributions modelling the fitness of a substitute to the context and to the target. The proximity of each candidate to the target word is computed as the inner product between the respective embeddings, followed by a softmax to get a probability distribution.

We modified the post-processing of the original system to exclude the candidate lemmatization and get inflected suggestions, rather than lemmas. We kept the lowercase post-processor followed by target exclusion which uses lemmatization to detect and exclude all forms of the target word. Finally, we increased the number of suggestions to 20 which we found increased the chances of finding a suitable simpler substitution candidate.

2.2 Candidate Ranking

We selected and ranked candidates based on a combination of their grammaticality, meaning preservation and simplicity scores (for which we provide detailed descriptions further down in this section). Despite a large number of metrics aiming to evaluate one or more aspects of a simplification at a time (BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), SARI (Xu et al., 2016), SAMSA (Sulem et al., 2018), Simple-QE (Kriz et al., 2020), ISiM (Mucida et al., 2022), Flesch Reading Ease Score (Kincaid et al., 1975), BERTScore (Zhang* et al., 2020), language model perplexity, etc.), not a single one of them excels at accurately measuring all three while also being publicly available. To rank the substitution candidates we thus chose to evaluate each aspect separately and to combine the scores through a simple heuristic giving twice as much weight to the simplicity score.

$$rank\ w_{n\leq N} = G_w \times (S_w \times 2 + M_w)$$

The rank of each substitution w_n of the $N = 20$ generated candidates is calculated as a function of its grammaticality $G \in \{0, 1\}$, simplicity $S \in [1, 6]$ and meaning preservation $M \in [1, N]$ scores. The top 10 ranking candidates (or less) are those included in the submission.

Grammaticality To evaluate the grammaticality of a sentence given a substitute candidate, we compare the coarse-grained part-of-speech (POS) and morphological features of both complex word and candidate in context. We use spaCy [1] to tokenize and parse the sentence, making sure not to split hyphenated complex words, since LexSubGen does not support multi-word expressions. We assign a score of 1 to all candidates whose features (person, number, mood, tense, etc.) correspond to those of the target word and 0 otherwise.

Meaning Preservation To evaluate the effect of a substitution candidate on the meaning of the original sentence we compute the similarity of the two sentences as a sum of the cosine similarities between their tokens' embeddings using BERTScore (Zhang* et al., 2020). The higher the similarity

[1] https://spacy.io/ | v. 3.1.3 | en-core-web-lg

between source and target sentences, the higher the chances that the substitution candidate's meaning is close to the one of the complex word. Candidates are ranked by decreasing $F1$ score with the best candidate receiving a score of 1 and the last one - a score equal to N.

Simplicity Arguably the most important aspect to evaluate of a given substitution candidate is whether or not it is simpler than the original complex word. Many synonyms a system (or even annotators) suggests may very well be grammatical, but if they do not simplify the concept within an acceptable degree of semantic variability, they fail to render the phrase easier to understand. The metric often times employed as a proxy for complexity is frequency, but frequency alone does not explain all the variation in lexical complexity datasets.

Our main contribution to this LS system is a more accurate measure of lexical complexity, notably a CEFR[2] vocabulary classifier, which we use to assign a complexity level to each substitution candidate. The lower the difficulty level, the higher a word's final rank.

The classifier is trained on data from the English Vocabulary Profile [3] (EVP) (Capel, 2012, 2015), a rich resource in British and American English which associates single words, phrasal verbs, phrases, and idioms not only with a CEFR level but with part of speech tags, definitions, dictionary examples and examples from learner essays. The corpus also contains distinct entries for distinct meanings of polysemous words, each associated with its own difficulty level. For example, we find 10 entries for the word form *run* in the American English section of the corpus, two noun forms and eight verbs, whose difficulty varies between A1 (*He can **run** very fast.*) and C2 (*He would like to **run** for mayor.*)

Rather than representing the vocabulary items by their frequency and/or surface-level characteristics (number of characters, number of syllables, etc.), we extract a semantic, contextual, dense vector representation of each item from a pre-trained masked language model [4] (Devlin et al., 2018) by first encoding the target word or MWE in context (using the dictionary and learner examples) and then aggregating all 12 hidden layers for all WordPieces. This representation of the dataset is then used to train a support vector classifier [5] (Platt et al., 1999).

The resulting model is able to assign a difficulty level between $1 \equiv A1$ and $6 \equiv C2$ to the meaning of any word or MWE as determined by its context.

3 Results and Discussion

We submitted results from two runs of the same system with the only difference being the grammaticality score. In our second submission, we disabled the filtering of morphologically inconsistent substitution candidates (in other words, we assign G_w a score of 1 for all w) after noticing that in some cases, some very appropriate candidates get filtered out following an erroneous morphological analysis. Both submissions achieve very similar results (Saggion et al., 2022), but the second one improves on the first on all but two metrics: ACC@1@Top1 and ACC@3@Top1 despite generating inappropriate candidates (both semantically and syntactically) in some rare cases.

Team	**ACC@1**	**MAP@3**	**Pot@3**
UniHD	0.809	0.583	0.962
UniHD	0.772	0.509	0.89
MANTIS	0.656	0.473	0.876
UoM&MMU	0.635	0.424	0.873
LSBert-baseline	0.597	0.407	0.823
RCML	0.544	0.382	**0.831**
RCML	0.541	0.371	0.801
GMU-WLV	0.517	0.352	0.753

Table 1: Top of the leaderboard for the English track

RCML outperforms the state-of-the-art LSBert baseline on 27 out of the total 51 metrics (including Precision and Recall). Table 1 shows the top of the leaderboard including our team's two submissions. RCML has Potential@3 of 0.831 which is higher than LSBert's (Qiang et al., 2020) and comparable to the top-scoring systems. This result suggests a potential for our system to assist human editors in the task of lexical simplification by proposing a few simpler synonyms to choose from. The system's Accuracy@1@top1 doubles when $K = 3$ which means that in 46% of the time, the most commonly suggested substitute is among our top 3 predictions.

[2] The Common European Framework of Reference for Languages (CEFR) organizes language proficiency in six levels, A1 to C2.

[3] https://www.englishprofile.org/american-english

[4] https://huggingface.co/bert-base-uncased

[5] sklearn.svm.SVC

Sentence	Gold@1	System@1
Putin was expected to formally register later in the day to run for president, [...] a period in which he grew more **authoritarian**.	dictatorial	nationalist
In Japan, rice with azuki beans [...] is traditionally cooked for **auspicious** occasions.	favorable	important
Police are appealing for information about anyone seen acting **suspiciously** lately at Bidston Hill, Bidston, to come forward.	doubtfully	strangely
And in the capital Damascus, regime forces raided [...] while **snipers** were stationed on the roofs of some buildings.	sharpshooters	guns

Table 2: Sentences with complex gold substitution candidates

Sentence	Gold@1	System@1
It **decomposes** to arsenic trioxide, elemental arsenic and iodine when heated in air at 200°C.	decays	changes
Lebanon is sharply split along **sectarian** lines, with 18 religious sects.	divided	religious
The stretch of DNA transcribed into an RNA molecule is called a transcription unit and **encodes** at least one gene .	encrypts	codes
Obama earlier dropped from night skies into Kabul [...], **cementing** 10 years of U.S. aid for Afghanistan after NATO combat troops leave in 2014.	bonding	securing

Table 3: Sentences with erroneous gold substitution candidates

3.1 Error Analysis

We analyzed manually the first 100 sentences of the test set, comparing the most popular substitution candidate among annotators with the most probable candidate suggested by RCML. We identified 15 sentences for which the best Gold annotation is more complex than the system's top candidate, vs. 3 cases where the roles are reversed. Table 2 provides a few examples of our tentative observations. Admittedly, choosing a lexical simplification candidate requires to strike a balance between simplicity and synonymy, but we would argue that simplicity should be the guiding factor.

In a number of cases (six in the gold annotations and ten in the system predictions) the top substitution candidate is semantically and/or morphologically incoherent. Table 3 lists some of the cases where we believe the annotators confused the meaning of the target complex word, while RCML provided a suitable candidate.

The error analysis of RCML allowed us to notice that its current version does not exclude or penalize candidates introducing repetitions in the sentence, while human annotators avoid those naturally.

Another examined sentence illustrates well the limits of distributional semantics and the pitfalls of structural ambiguity. The complex word in the sentence below is a predicate whose argument is the noun *fighters*, but RCML first suggests predicates compatible with *explosives — hidden, positioned, stored*.

> The unsophisticated nature of the attack suggests little planning beyond having fighters and some explosives **prepositioned** in the vicinity of Kabul.

4 Conclusion

In this paper, we describe a modular lexical simplification system for English which requires no training data. RCML uses LexSubGen to generate substitution candidates before evaluating their grammaticality, meaning and simplicity. The latter is predicted by a 6-class contextual CEFR vocabulary classifier. The system is easily adaptable to other languages provided a trained CEFR vocabulary classifier in the languages in question. It also has the capacity to perform personalized lexical simplification, a particularly relevant approach when simplifying text for language learners at different proficiency levels.

References

Nikolay Arefyev, Boris Sheludko, Alexander Podolskiy, and Alexander Panchenko. 2020. Always keep your target in mind: Studying semantics and improving performance of neural lexical substitution. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1242–1255, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Annette Capel. 2012. Completing the english vocabulary profile: C1 and c2 vocabulary. *English Profile Journal*, 3.

Annette Capel. 2015. The english vocabulary profile. *English profile in practice*, 5:9–27.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Reno Kriz, Marianna Apidianaki, and Chris Callison-Burch. 2020. Simple-qe: Better automatic quality estimation for text simplification. *arXiv preprint arXiv:2012.12382*.

Lucas Mucida, Alcione Oliveira, and Maurilio Possi. 2022. A language-independent metric for measuring text simplification that does not require a parallel corpus. In *The International FLAIRS Conference Proceedings*, volume 35.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

John Platt et al. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2020. Lexical simplification with pretrained encoders. *AAAI*, 34(05):8649–8656.

Horacio Saggion, Sanja Stajner, Daniel Ferres, Kim Cheng Sheang, Matthew Shardlow, Kai North, and Marcos Zampieri. 2022. Findings of the tsar-2022 shared task on multilingual lexical simplification. In *Proceedings of TSAR workshop held in conjunction with EMNLP 2022*.

Sanja Stajner, Daniel Ferres, Matthew Shardlow, Kai North, Marcos Zampieri, and Horacio Saggion. 2022. Lexical simplification benchmarks for English, Portuguese, and Spanish. *Frontiers in Artificial Intelligence*, 5.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. Semantic structural evaluation for text simplification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 685–696, New Orleans, Louisiana. Association for Computational Linguistics.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.