Multi-Word Lexical Simplification

Piotr Przybyła
Institute of Computer Science,
Polish Academy of Sciences
Warsaw, Poland
piotr.przybyla@ipipan.waw.pl

Matthew Shardlow
Department of Computing and Mathematics
Manchester Metropolitan University
Manchester, UK
M.Shardlow@mmu.ac.uk

Abstract

In this work we propose the task of *multi-word lexical simplification*, in which a sentence in natural language is made easier to understand by replacing its fragment with a simpler alternative, both of which can consist of many words. In order to explore this new direction, we contribute a corpus (MWLS1), including 1462 sentences in English from various sources with 7059 simplifications provided by human annotators. We also propose an automatic solution (Plainifier) based on a purpose-trained neural language model and evaluate its performance, comparing to human and resource-based baselines.

1 Introduction

Text simplification is the task of automatically modifying natural language text to improve its ease of understanding, whilst preserving the overall meaning. The diverse applications range from second language learners to lay readers of scientific texts to stroke victims. This challenge is often defined in the framework of *lexical* text simplification (LS), where individual words are replaced with their simpler equivalents (Paetzold and Specia, 2017b).

However, single-word substitutions do not cover the full complexity of techniques humans use to approach text simplification, including replacements, deletions, addition and sentence splits. They could be modelled implicitly, by sentence-to-sentence monolingual translation approaches (Zhu et al., 2010; Zhang and Lapata, 2017), or explicitly, by collecting data (Alva-Manchego et al., 2020) and developing methods (Dong et al., 2019) focusing on particular types of transformations. With a similar motivation, we propose in this work a new task going beyond single word substitution, namely *Multi-Word Lexical Simplification* (MWLS).

The aim of MWLS is to replace a given fragment (short sequence of words) with its simpler version, so that the enclosing sentence retains its meaning, but becomes easier to understand. See table 1 for several examples from our corpus. Note how these fragment-to-fragment replacements can involve words substitutions, expansions, deletions and other restructuring.

This paper explores the task of MWLS in two directions. Firstly, we contribute a dataset of 1462 sentences with 7059 simplifications obtained through crowdsourcing. Secondly, we design a method for generating such simplifications automatically. Our solution, called the *Plainifier*, is inspired by a recently proposed method for LS utilising language models (Qiang et al., 2020), which we extend so that multi-word simplifications can be obtained. In order to encourage more research on the problem, we make the dataset[1], the language model[2] and the Plainifer code[3] openly available.

2 Background

Typical lexical simplification systems have followed a four stage pipeline of complex word identification (Shardlow, 2013), substitution generation, word sense disambiguation and synonym ranking (Paetzold

[1] https://github.com/piotrmp/mwls1
[2] https://github.com/piotrmp/tersebert
[3] https://github.com/piotrmp/plainifier

and Specia, 2017a). However, in the past these systems have mostly focused on identifying and replacing single words (Shardlow, 2014; Paetzold and Specia, 2016c).

In the recent shared task on complex word identification (Yimam et al., 2018) the English portion of the dataset (Yimam et al., 2017) contained a proportion (26%) of multi-word expressions (MWEs). This presented a challenge to the participants who needed to adapt their systems to identify these cases. Whilst some did not provide any specific treatment of MWEs, a common technique was to average features of the sub-words in an MWE to give an overall feature set (Alfter and Pilán, 2018; Hartmann and Dos Santos, 2018). The winning submission however (Gooding and Kochmar, 2018) used a "greedy" approach whereby they assigned all MWEs to the complex class. Clearly, MWEs deserve further attention in the lexical simplification world.

There are a few prior attempts to integrate MWEs into lexical simplification such as the use of decompositional rules to identify key words in MWEs that can then be used for ranking (Amoia and Romanelli, 2012). In other languages which involve heavy compounding, such as Swedish, lexical simplification is multi-word by nature. Lexical complexity can be assessed by identifying relevant lexical substrings in a target word and using these to compute features (Abrahamsson et al., 2014). More recently, RecLS (Gooding and Kochmar, 2019b) was developed to perform lexical simplification recursively. If 2 consecutive words are marked as complex by the algorithm then they will be simplified one at a time, with the algorithm deciding whether to stop at each iteration. Although this does allow the handling of MWEs, it does not allow for the generation of phrases to replace the MWE, instead simplifying each word in place.

Sentence simplification (Nisioi et al., 2017) is a combination of syntactic and lexical simplification. Typically, it is accomplished using techniques from machine translation. The baseline dataset for this form of simplification is SARI (Xu et al., 2016), which comprises target sentences and their reference simplifications. It is worth noting however, that many transformations are in fact multi-word lexical simplifications as we are trying to produce here, or combinations thereof.

To the best of our knowledge, no corpora or methodology has been previously published on simplifying text in English by replacing multi-word fragments, which is the focus of this study.

3 Task and Dataset

The goal of multi-word lexical simplification is to simplify a given sentence in a natural language by replacing its fragment (a sequence consisting of one or several words) with another fragment, such that:

- the new sentence is a correct sentence in the language,
- the overall meaning of the sentence is preserved,
- the new sentence is simpler (i.e. easier to understand) than the original.

Note that we impose no restrictions on the grammatical roles of words included in the fragment, e.g. it does not have to constitute a multi-word expression. Moreover, while it would be desirable for the fragment being replaced to be the hardest part in the sentence, we do not consider selecting such a fragment part of the MWLS task, since it belongs to a problem complex word identification (CWI), which has recently been explored in a multi-word setting (Kochmar et al., 2020).

Given that this problem formulation is novel and no resources for it are available, we have decided to prepare a new dataset that would include a large number of such multi-word simplifications provided by human annotators. The rest of this chapter shows how we (a) collect sentences from corpora with complex language, (b) use a CWI solution to select fragments worth simplifying and (c) obtain manual replacements from crowdsourcing workers. Finally, we briefly describe the characteristics of the obtained dataset.

3.1 Sentence Selection

In order to obtain a collection of sentences in English that will be simplified, we choose the same three sources that were employed in a recent study on CWI (Shardlow et al., 2020), namely:

- BIBLE: World English Bible translation from a parallel corpus (Christodouloupoulos and Steedman, 2015),

Make a given sentence simpler (easier to read and understand) by replacing the highlighted fragment using **1, 2 or 3 words**. Enter your replacement fragment (NOT a whole sentence) in the input field of the form.
The modified sentence should:

1. be more easily understandable to less proficient (for example, non-native) speakers of English,
2. keep the general meaning of the original,
3. remain a correct sentence in English and 'read well'.

If you can't understand the original sentence or find a good enough replacement, use a question mark **'?'** instead.

Please **do not**:

- provide more than one replacement for a fragment,
- use more than 3 or less than 1 words,
- overuse the question mark when a reasonable replacement is possible,
- simply copy the highlighted fragment.

You **can**:

- freely choose a replacement when more than one is possible, for example: *attire* → *dress*, *attire* → *outfit*, *attire* → *clothes*,
- change the grammatical structure of the sentence, for example: *is provided with* → *gets*,
- use a replacement even when its meaning is not exactly the same, but just close enough, for example: *stimulate* → *help*, *numerous sins* → *many crimes*,
- use a replacement with a different number of words than original, for example *drink excessively* → *overdrink*, *drink excessively* → *drink too much*.

The results will be evaluated manually.

Figure 1: Instructions used by annotators providing replacements for the MWLS1 dataset.

- EUROPARL: English text from the European Parliament proceedings compiled as the Europarl corpus (Koehn, 2005),
- BIOMED: Text of biomedical publications gathered in the CRAFT corpus (Bada et al., 2012).

The three sources vary greatly in the content and language style, yet they all provide enough complexity to justify simplification efforts.

From each of the three corpora above, we randomly select 10,000 sentences (verses in case of BIBLE) and apply the neural CWI model published by Gooding and Kochmar (2019a) to assess the complexity of individual words. Next, for each of the corpora and allowed fragment lengths (1, 2 or 3 words) we select 1,000 sentences including fragments with the highest complexity score. In the process we ensure that there are no duplicate sentences and that fragments do not cover proper names or very rare words, i.e. occurring in less than 2% of documents from the Google Books corpus[4]. This helps to avoid including terms that require expert knowledge to simplify (e.g. *diacylglycerol*).

3.2 Crowdsourcing

In the crowdsourcing phase, Amazon Mechanical Turk (MTurk) is used to obtain simplifications for the selected fragments. In order to obtain a balanced dataset, each batch contains the same number of sentences (tasks) from each source and fragments of each length. The MTurk workers are provided with instructions (Figure 1), asking them to provide replacements of up to 3 words for the highlighted fragment. They can also input a question mark, if they are unable to understand the sentence or find a good simplification.

Each task is given to 5 workers in parallel. Since MTurk does not allow to select workers based on their native language or fluency in English, the task is available only to those from English-speaking countries that either possess a *Masters* qualification or have completed at least 1,000 tasks with 98% acceptance rate. To ensure the quality of our dataset remains high, we evaluate the replacements manually by verifying their compliance with the provided instruction. In total, 4.49% of replacements were rejected and other workers were assigned to these tasks according to the rules of MTurk. The most common reason for rejection (42% of cases) was using more than three words in a replacement. Workers with less than 97% acceptance rate in our task are excluded from future batches.

[4] http://storage.googleapis.com/books/ngrams/books/datasetsv2.html

3.3 MWLS1 Dataset

Case ID	Source	Sentence	Replacement
CASE_7739	BIOMED	The main difference in the two lines **essentially resides** in the strength of the promoter.	is basically
CASE_241	BIBLE	Thus says Yahweh of Armies, "They **shall thoroughly glean** the remnant of Israel. Turn again your hand as a grape gatherer into the baskets."	will gather
CASE_5327	EUROPARL	I support Ms Lulling's recommendations that the national systems should recognise the importance of protecting self-employed workers, and we should stand against all forms of **discrimination**, but I am still not convinced that this House is best placed to work on employment matters.	bias and unfairness
CASE_6461	BIOMED	Other **potentially biologically relevant** substrates include cholecystokinin and possibly other neuropeptides [21].	relevant
CASE_2260	BIBLE	A man's **foes** will be those of his own household.	enemies

Table 1: Five examples of sentences from the MWLS1 dataset, each shown with its identifier, source corpus, highlighted fragment to be simplified and one of the replacements provided by annotators.

The obtained dataset, called MWLS1 (Multi-Word Lexical Simplification 1) contains 1462 sentences with 7059 simplifications. The fraction of instances where workers were unable to provide a simplification is 3.43%. Examples of the types of simplifications that are present in our dataset are shown in Table 1. It is clear that the types of simplifications are diverse. Sometimes reducing (CASE_241, CASE_6461) and sometimes expanding on (CASE_5327) the target phrase. The substitutions preserve the meaning of the original target, whilst attempting to rephrase in a simpler form. The simplifications where both the replaced and replacement text consist of one word, fitting the traditional lexical simplification framework, account for 29% of the dataset.

We have provided comprehensive statistics on our dataset in Table 2. In this table, we have split the analysis into each source corpus and length of the replaced fragment (n-gram). We report the number of instances per section, demonstrating that our corpus contains an even distribution of genres and n-gram lengths. The mean number of replacements per sub-section is reported, showing that all subsets received between 4 and 5 replacements per target on average. The number of complete instances refers to those that received all 5 replacements. An exceptional statistic in this is the Biomedical trigrams, of which only 57.42% received a full complement of 5 answers. This likely implies that these were particularly difficult or unknown to the annotators.

The number of words given as an answer per target roughly tracks the number of words in the target. For unigrams and bigrams it is consistently higher, indicating a tendency to expand when explaining. For trigrams, it is consistently lower. However this is to be expected as answers were limited to 3 tokens. This is a limitation of our approach and a wider study may find that trigrams are also typically expanded, if annotators are given the choice to do so.

We compute the agreement of annotators using a custom metric. We do not expect annotators to agree. In fact, the corpus would no longer capture the diverse possibilities of each simplification case if annotators consistently gave the same response. However, it is still interesting to identify cases where annotators did give the same answers as this indicates that there is some coincidental agreement. We have calculated this agreement by identifying for each set of answers how many annotators gave the most common answer. The overall agreement score is then the mean of these values. In this scheme, an agreement of 1 would indicate that no annotators agreed with each other, whereas an agreement of 5 would indicate that all annotators agreed on the replacement all the time. The agreement is consistently around 2 for unigrams, but lower for bigrams and trigrams, indicating that annotators were more likely to come up with the same answer for a shorter target phrase. There were 10 sentences in our corpus where all five annotators gave the same response, all unigrams and split across the three genres. The occurrence of these cases is an interesting phenomenon and warrants further investigation.

The dataset is released under a CC BY-NC-SA 4.0 Licence, which preserves the licence conditions of the source corpora.

Subset	Sentences	Replacements	Complete	Words	Agreement
BIBLE 1	168	4.9345	95.23%	1.2393	2.1548
BIBLE 2	166	4.8675	89.76%	2.0169	1.3373
BIBLE 3	160	4.9125	92.50%	2.7025	1.2625
EUROPARL 1	166	4.9699	98.19%	1.2614	2.1386
EUROPARL 2	158	4.9241	94.30%	2.0418	1.2025
EUROPARL 3	161	4.8447	87.58%	2.6671	1.0994
BIOMED 1	166	4.8494	87.35%	1.2024	2.0542
BIOMED 2	162	4.7531	82.10%	1.9741	1.2654
BIOMED 3	155	4.3742	57.42%	2.3652	1.0581
BIBLE All	494	4.9049	92.5%	1.9745	1.5911
EUROPARL All	485	4.9134	93.4%	1.9823	1.4887
BIOMED All	483	4.6646	75.98%	1.8344	1.47
All 1	500	4.918	93.60%	1.2344	2.116
All 2	486	4.8477	88.68%	2.0107	1.2695
All 3	476	4.7143	79.41%	2.5807	1.1408
All	1462	4.8283	87.34%	1.9308	1.5171

Table 2: Statistics on our corpus, divided into *subsets* of sentences, according the source corpus (Bible, Europarl, Biomed or All) and the number of words in the replaced fragment (1, 2, 3 or All). We show the number of *sentences* in each subset, the mean number of *replacements* given for the instances in the category and how many instances are *complete*, i.e. when all 5 answers were given. *Words* shows the mean number of words in provided replacements for the given target words. *Agreement* indicates how many annotators agreed on a replacement on average.

4 Simplification Method

Our approach to the MWLS problem, called the *Plainifier*, is an extension of the unsupervised method for single word lexical simplification by Qiang et al. (2020). Following their work, we generate candidate replacements using BERT predictions for a given context and rank them according to language-model probability, simplicity and similarity of meaning to the original text. Nevertheless, there are important differences caused by the fact that both replaced and replacement text can be of any length. Specifically, the candidate generation procedure (section 4.2) is a multi-step recursive procedure, which requires a specially trained version of BERT (section 4.1). Also, during candidate ranking (section 4.3), our system needs to compare the quality of the generated fragments even though they differ in length.

Note that Plainifier does not perform CWI and thus its input consists of the the full sentence text and coordinates for the fragment that should be simplified.

4.1 TerseBERT

When BERT is used as a language model (Devlin et al., 2018), it estimates $P(\langle \mathbf{c}_l, t^*, \mathbf{c}_r \rangle)$, i.e. the likelihood of a token[5] t^* occurrence given its context on the left $\mathbf{c}_l = \langle \ldots, c_{-2}, c_{-1} \rangle$ and right $\mathbf{c}_r = \langle c_1, c_2, \ldots \rangle$[6]. However, the candidate generation procedure of Plainifier (section 4.2) has to recognise the situations where no tokens are needed in a given context. In other words, we want to measure $P(\langle \mathbf{c}_l, \mathbf{c}_r \rangle)$: the likelihood with which the contexts $\mathbf{c}_l$ and $\mathbf{c}_r$ follow each other directly rather than with a token between them.

In order to make BERT capable of assessing this quantity, we take a pretrained model (BERT-Large, Uncased, WWM) and resume training with a certain modification. Namely, one third of the `[MASK]` elements cover a special token, denoted as `[NONE]`, inserted randomly between tokens of the original sentence. As a result, BERT learns to assign a high score to a `[NONE]` token when a given context needs no additional words and this score is used as an estimate for $P(\langle \mathbf{c}_l, \mathbf{c}_r \rangle)$. We run the additional training

[5] *Token* refers to WordPiece tokenisation, which is used by BERT. Longer words can be represented using several tokens.

[6] Angle brackets $\langle\rangle$ denote token sequences.

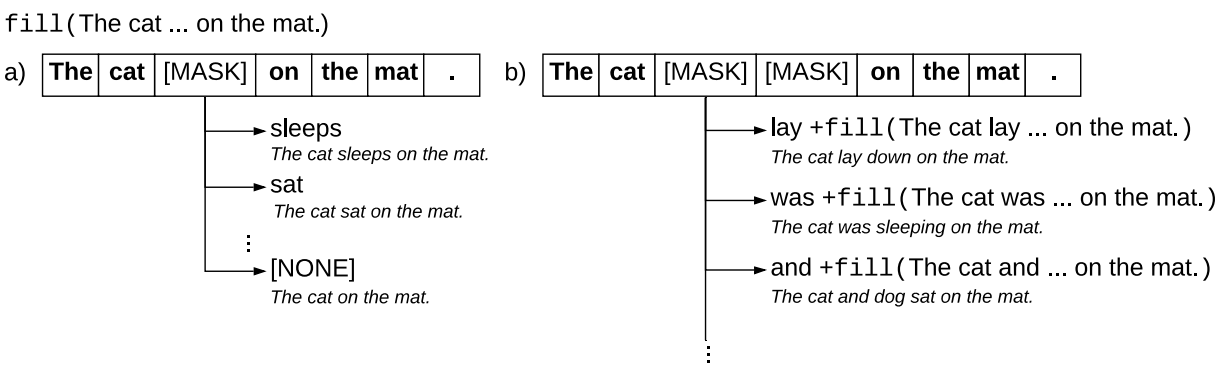

Figure 2: Outline of the candidate generation procedure. See description in text.

process for 5000 training steps, 128 sentences each, using text extracted from English Wikipedia, which was also used in the original training. We refer to the resulting model as *TerseBERT*.

4.2 Candidate generation

Generation of replacement candidates is performed in a two-step recursive procedure, visualised in Figure 2 on an example of replacing *sat* from *The cat sat on the mat*. In step (a), a gap created by removing the replaced token is filled with a single `[MASK]`, for which the predictions are acquired from TerseBERT. This method is used by Qiang et al. (2020) to obtain one-word candidates, such as *The cat sleeps on the mat*, and their likelihoods. The multi-words setting in Plainifer requires step (b), in which the gap is filled with two `[MASK]` elements and the best (according to ranking described in section 4.3) K predictions for the first position are obtained. For each of such prefixes, e.g. *was*, the procedure is executed recursively with the context extended accordingly, e.g. *The cat was ... on the mat*, and K replaced by $\frac{K}{2}$. The process of assembling a candidate continues until either a maximum length L is reached or the probability of a `[NONE]` token exceeds the threshold p, indicating no more words in a gap are necessary. The parameters K, L and p determine how extensive (and computationally expensive) the search process is. In our experiments we use $K = 16$, $L = 3$ (or length of the replaced fragment, if higher) and $p = 0.5$.

In order to broaden the candidate search, two modifications are made. Firstly, the candidate generation is executed both forwards (as shown in figure 2) and backwards (generating candidate from its last token) and the resulting lists are combined. Secondly, we guarantee that when choosing prefixes at a given position, the original token at that position is also included. This ensures that when looking for replacements for *sleeps soundly*, fragments such as *sleeps well* are considered, even if the likelihood for *sleeps* appears low.

4.3 Candidate ranking

All candidate fragments generated in the previous step are assessed in terms of *probability*, *similarity* and *familiarity*. Each of these is a number in (0,1):

- *Probability* (P) expresses how likely a fragment is in the context according to the language model. This number could be directly obtained from TerseBERT for each token using a softmax transformation. The fragment-level probability is computed as a product over probabilities of tokens it consists of.
- *Similarity* (S) measures how much the meaning of the candidate resembles the original fragment, regardless of the context. The similarity between tokens $\mathrm{sim}(t_1, t_2)$ is computed as a cosine similarity between *fastText* (Mikolov et al., 2017) representations. The similarity between fragments is computed by finding the best alignment of their tokens:

$$\mathrm{sim}(\langle t_1, t_2, \ldots, t_{n_1}\rangle, \langle u_1, u_2, \ldots, u_{n_2}\rangle) = \frac{1}{n_1 + n_2}\left[\sum_{i=1}^{n_1} \max_j \mathrm{sim}(t_i, u_j) + \sum_{j=1}^{n_2} \max_i \mathrm{sim}(t_i, u_j)\right]$$

- *Familiarity* (F) is intended to capture how likely a token is to be known by a reader by measuring its frequency in language. It is computed as a number of documents it occurred in according to *Google Books Ngrams*[7], scaled to (0,1). Familiarity of the fragment is obtained by taking the minimum of values assigned to included tokens.

Similar quantities are also taken into account in the LS solution (Qiang et al., 2020), but our formulation allows them to be computed at the level of multi-token fragments. Moreover, the final score of a chunk c is computed as a product $score(c) = [P(c)]^{\alpha_1} \times [S(c)]^{\alpha_2} \times [F(c)]^{\alpha_3}$. We introduce the parameters $\alpha_1, \alpha_2, \alpha_3$ in Plainifier for the final score to better reflect the quality of candidates. Their values could be equal (by default) or adjusted using a small tuning data portion (see section 5.2).

5 Evaluation

The evaluation of Plainifier is performed by comparing its output to the replacements provided by humans in the crowdsourcing process. We only use the 1277 sentences, which have all 5 replacements. Given that our method does not need training data, but offers possibility of tuning, we randomly select 100 sentences for this purpose, leaving the remaining 1177 for computing evaluation metrics.

5.1 Measures

In the basic evaluation scenario we cast our problem as an information retrieval task, treating the candidates ordered by decreasing score as a ranking list and the humans' replacements as relevant results. We compute precision at 5, reflecting the number of relevant results in top 5 positions, and NDCG, taking into account the whole list (Järvelin and Kekäläinen, 2002). We also compute *potential*, which was introduced in the simplification context by Paetzold and Specia (2016b) and measures in how many cases at least one of the expected simplifications was present among the generated candidates.

Note that these measures are overly pessimistic, as they assume any replacement not provided by humans is wrong. In fact there can be many simplifications for a given fragment not covered by these five responses. For example, the gold answers for CASE_241 in Table 2 may contain the following 'will gather', 'shall harvest' and 'will glean'. In a strict evaluation, only these phrases would be accepted, however a system may produce other acceptable combinations of these words such as 'will harvest', 'shall glean' or 'shall gather', none of which would be accepted by the strict metrics.

In response to this, we develop a new evaluation metric, called BOW@K. In this metric, we take the top-K responses given by a system and create a bag of words from these (i.e., a set of all the distinct words). For each answer we then calculate the percentage of words in the answer that can be found in the bag of words. The score for the instance is the mean overlap of the five answers. The instance scores are averaged to give a mean average overlap for the dataset. Using this new metric, the answers above would now be accepted. Clearly this is more lenient than strict matching, and may be open to abuse by systems that optimise to it. However, similar to BLEU or ROUGE score, the metric is useful to understand how different systems have performed on our dataset whilst allowing some leniency for the wide problem space in which we are working.

5.2 Tuning

In order to tune the $\alpha_1, \alpha_2, \alpha_3$ parameters, we first run the Plainifier using the default setting ($\alpha_1 = \alpha_2 = \alpha_3$), re-rank the candidates according to modified values and measure the NDCG for the new list. We try the 55 combinations of $\alpha_1 = \frac{i}{9}, \alpha_2 = \frac{j}{9}, \alpha_3 = \frac{k}{9}$, where $i, j, k \in \{0 \dots 9\}, i + j + k = 9$.

Figure 3 shows the results of tuning in a ternary plot. The best NDCG is achieved for $\alpha_1 = \frac{1}{9}, \alpha_2 = \frac{6}{9}, \alpha_3 = \frac{2}{9}$ (white diamond), i.e. the signal coming from BERT is dampened, while the similarity metric has higher priority. The evaluation includes both the default and tuned version of the method.

5.3 Baselines

To put our results in perspective, we also evaluate a human baseline solution. For each replacement provided by annotators for a given sentence, we compute the evaluation measures by treating this re-

[7] http://storage.googleapis.com/books/ngrams/books/datasetsv2.html

placement as a one-element list of results and the remaining replacements as the relevant results. These values, averaged over all replacements in all sentences, give us an indication of how good humans are at predicting each other's simplifications. Note that Prec@5 and Potential are equal in this scenario.

We also include two further baselines to evaluate our system against. The first baseline leverages SimplePPDB (Pavlick and Callison-Burch, 2016), which is a filtered version of PPDB (Ganitkevitch et al., 2013). For each target (replaced fragment), we first search in Simple PPDB and if it is found, return all the paraphrases associated with it. If the target is not found, the baseline attempts to find a replacement for each constituent word. If M replacements were found for the first token and N replacements found for the second, these are combined to give $M \times N$ new candidate substitutions. SimplePPDB contains a simplification score for each paraphrase, which is used to rank the resulting candidates (values are multiplied upon combining candidates). Our Wordnet baseline identifies the synonyms of a target word in Wordnet (Miller, 1998) and combines these using the same logic as for SimplePPDB. We used the Google Web1T unigrams to rank the resulting candidates.

6 Results

The results of the evaluation of Plainifier and the baselines with respect to subsets of MWLS1 are shown in table 3. We can see that the performance of Plainifier is much better than any of the automatic baselines in every sentence subset and evaluation measure, e.g. obtaining NDCG=0.1396 on all sentences compared to 0.0388 of SimplePPDB or 0.0279 of WordNet. The automatic solution is however still far from human performance, which could be seen in the Prec@5 results, where the precision of the top 5 candidates from the Plainifier is much lower than accuracy of individual replacements provided by human annotators. The high values of Potential of Plainifier (100% in BIOMED 1) indicate that the correct fragments are available among the candidates, but their ranking could be improved. The tuned version yields consistently better results (or equally as good in a few cases) than the default.

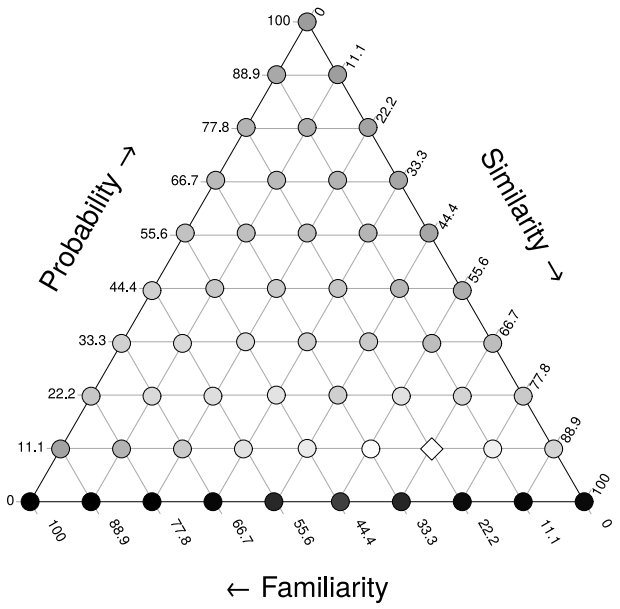

Figure 3: Ternary plot showing the results of tuning the parameters α_1, α_2, α_3, reflecting the importance of probability, similarity and familiarity, respectively. The greyscale indicates NDCG values: from 0.0465 (black) to 0.1576 (white).

We can see that all three source corpora provide challenges of a similar complexity level to the automatic solutions, with the results ordering dependent on the selected measure. Interestingly, it is not the case for the human baseline, which achieves consistently low results on EUROPARL sentences. The influence of original fragment length is, on the other hand, clear: in every source corpus, predicting replacements for longer fragments is much harder. This holds for both automatic solutions and human baselines, in correspondence with the number of completed answers visible in table 2.

Additionally, note that the values of NDCG and Prec@5, which could theoretically reach 1.0, remain much lower than that, peaking at 0.2756 for tuned Plainifier on EUROPARL 1 subset. This shows how challenging the MWLS task is, both in terms of automatically generating good-quality simplifications and evaluating them, taking into account numerous possibly correct answers.

Finally, we report BOW@K, where $K = 5$. We chose this value of K to make the results comparable with Precision@5. It can be seen that the results have improved for all systems by allowing the lenient BOW matching. The Human baseline still outperforms the Plainifier for this metric, as for Prec@5, however the results of the Plainifier are now closer to the baseline. If we increase K the metric also

Metric	Subset	Plainifier		Baselines		
		Default	Tuned	Human	SimplePPDB	WordNet
NDCG	BIBLE all	0.1505	**0.1528**	0.1116	0.0307	0.0293
	EUROPARL all	0.1453	**0.1575**	0.0999	0.0453	0.0267
	BIOMED all	0.1286	**0.1460**	0.1065	0.0401	0.0310
	All 1	0.2475	**0.2578**	0.2112	0.0973	0.0611
	All 2	0.0833	**0.0936**	0.0475	0.0059	0.0111
	All 3	0.0412	**0.0458**	0.0266	0.0036	0.0059
	All	0.1310	**0.1396**	0.1015	0.0388	0.0279
Prec@5	BIBLE all	0.0338	0.0343	**0.2174**	0.0127	0.0164
	EUROPARL all	0.0436	0.0465	**0.1913**	0.0237	0.0150
	BIOMED all	0.0323	0.0360	**0.2059**	0.0215	0.0204
	All 1	0.0721	0.0767	**0.3811**	0.0462	0.0351
	All 2	0.0170	0.0180	**0.1069**	0.0041	0.0072
	All 3	0.0056	0.0079	**0.0575**	0.0039	0.0028
	All	0.0338	0.0365	**0.1929**	0.0195	0.0161
Potential	BIBLE all	0.9225	**0.9366**	0.2174	0.1291	0.1150
	EUROPARL all	0.8668	**0.8765**	0.1913	0.2228	0.0969
	BIOMED all	0.8414	**0.8656**	0.2059	0.1855	0.0995
	All 1	0.9885	**0.9908**	0.3811	0.4296	0.1755
	All 2	0.8175	**0.8406**	0.1069	0.0386	0.0643
	All 3	0.0412	**0.0458**	0.0266	0.0197	0.0310
	All	0.6394	**0.6451**	0.0575	0.1767	0.0952
BOW@5	BIBLE all	0.1116	0.1243	**0.2085**	0.0674	0.0990
	EUROPARL all	0.1051	0.1140	**0.1607**	0.0925	0.0750
	BIOMED all	0.0927	0.1138	**0.1777**	0.0815	0.0779
	All 1	0.1364	0.1369	**0.1980**	0.0834	0.0598
	All 2	0.0853	0.1068	**0.1742**	0.0785	0.0979
	All 3	0.0821	0.1095	**0.2188**	0.1428	0.1789
	All	0.1031	0.1187	**0.1964**	0.0997	0.1083

Table 3: Values of evaluation measures computed on subsets of sentences coming from different corpora (BIBLE, EUROPARL and BIOMED) and with different lengths of replaced text (1, 2 or 3 words). We show the results of the Plainifier using default and tuned parameters and of three baselines.

improves with 0.1767 at $K = 10$, 0.2423 at $K = 20$ and 0.2808 at $K = 30$ (results on the tuned plainifier with the full dataset). The Plainifier also outperforms SimplePPDB and WordNet on these baselines.

7 Discussion

Our main motivation in this work was to improve the simplification process by introducing a task that goes beyond word-for-word substitution. The fact that we have been able to obtain replacements from annotators in a vast majority of cases suggests that MWLS is indeed a valid task for human simplification. Nevertheless, the cases with missing replacements indicate the possible limitations of this framework. Apart from understandable cases of insufficient knowledge (especially in the biomedical domain), there have been many cases of words, such as *circumcision*, for which no simplification within the allowed 3 words is possible and they could only be made understandable by providing a longer explanation. We think this phenomenon points to a potential for future work on the CWI task – namely, in differentiating complex words that could be simplified through substitution (e.g. *foes* to *enemies*) from those requiring explanation (e.g. *circumcision*).

Another challenging aspect of the task is evaluation: when so many substitutions are possible, a value provided may be valid even if it was not included in gold-standard human annotations. The situation is similar to other generative tasks, such as summarisation or translation, where relaxed matching measures were developed (e.g. BLEU). We have proposed the BOW@K metric in similar spirit, but more work is necessary to assess, to what degree they correlate with manual assessment of quality.

Regarding automatic simplification, while the Plainifier achieves better results than resource-based baselines, it leaves much room for improvement. The comparison of results in terms of Prec@5 and Potential shows that many relevant replacements are available in the candidate list, but the ranking method fails to assess them properly. Manual inspection shows that the most challenging aspect is preserving the

original meaning when computing the cosine of word vectors blends the semantic and stylistic similarity.

Our corpus is not user specific, although other studies have shown that user effects are present in simplification (Paetzold and Specia, 2016a). We would hope that this work fits into the larger picture of ongoing lexical simplification research to adapt simplification to further users, genres and problem-types.

8 Conclusion

In this article we have defined the task of multi-word lexical simplification and explored its nature while obtaining both human and automatic solutions to the problem. We hope that the resources we contribute, including the gathered data, developed code and trained models, will be of use for future researchers taking up this challenge.

Acknowledgements

This work was supported by the *Polish National Agency for Academic Exchange* through a *Polish Returns* grant number PPN/PPO/2018/1/00006 and a computing grant number 447 at the *Poznan Supercomputing and Networking Center*.

References

Emil Abrahamsson, Timothy Forni, Maria Skeppstedt, and Maria Kvist. 2014. Medical text simplification using synonym replacement: Adapting assessment of word difficulty to a compounding language. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 57–65, Gothenburg, Sweden. Association for Computational Linguistics.

David Alfter and Ildikó Pilán. 2018. SB@GU at the complex word identification 2018 shared task. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 315–321, New Orleans, Louisiana. Association for Computational Linguistics.

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020. ASSET: A Dataset for Tuning and Evaluation of Sentence Simplification Models with Multiple Rewriting Transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679. Association for Computational Linguistics.

Marilisa Amoia and Massimo Romanelli. 2012. SB: mmSystem - using decompositional semantics for lexical simplification. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 482–486, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William A. Baumgartner, K. B. Cohen, Karin Verspoor, Judith A. Blake, and Lawrence E. Hunter. 2012. Concept annotation in the CRAFT corpus. *BMC Bioinformatics*, 13(1).

Christos Christodouloupoulos and Mark Steedman. 2015. A massively parallel corpus: the Bible in 100 languages. *Language Resources and Evaluation*, 49(2):375–395.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186. Association for Computational Linguistics.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. EditNTS: An Neural Programmer-Interpreter Model for Sentence Simplification through Explicit Editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402, Florence, Italy. Association for Computational Linguistics.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. 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, Atlanta, Georgia. Association for Computational Linguistics.

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, New Orleans, Louisiana. Association for Computational Linguistics.

Sian Gooding and Ekaterina Kochmar. 2019a. Complex Word Identification as a Sequence Labelling Task. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1148–1153, Florence, Italy. Association for Computational Linguistics.

Sian Gooding and Ekaterina Kochmar. 2019b. 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, Hong Kong, China. Association for Computational Linguistics.

Nathan Hartmann and Leandro Borges Dos Santos. 2018. NILC at CWI 2018: Exploring feature engineering and feature learning. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 335–340, New Orleans, Louisiana. Association for Computational Linguistics.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446.

Ekaterina Kochmar, Sian Gooding, and Matthew Shardlow. 2020. Detecting Multiword Expression Type Helps Lexical Complexity Assessment. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4426–4435, Marseille, France. European Language Resources Association.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, pages 79–86.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2017. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

George A Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91, Vancouver, Canada. Association for Computational Linguistics.

Gustavo Paetzold and Lucia Specia. 2016a. Understanding the lexical simplification needs of non-native speakers of English. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 717–727, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Gustavo H. Paetzold and Lucia Specia. 2016b. Benchmarking Lexical Simplification systems. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3074–3080, Portorož, Slovenia. Association for Computational Linguistics.

Gustavo H Paetzold and Lucia Specia. 2016c. Unsupervised lexical simplification for non-native speakers. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Gustavo Paetzold and Lucia Specia. 2017a. Lexical simplification with neural ranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 34–40, Valencia, Spain. Association for Computational Linguistics.

Gustavo H. Paetzold and Lucia Specia. 2017b. A survey on lexical simplification. *Journal of Artificial Intelligence Research*, 60:549–593.

Ellie Pavlick and Chris 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, Berlin, Germany. Association for Computational Linguistics.

Jipeng Qiang, Yun Li, Yi Zhu, Yunhao Yuan, and Xindong Wu. 2020. Lexical Simplification with Pretrained Encoders. In *Proceedings of the AAAI Conference on Artificial Intelligence*. Association for the Advancement of Artificial Intelligence.

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. CompLex — A New Corpus for Lexical Complexity Predicition from Likert Scale Data. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*, pages 57–62, Marseille, France. European Language Resources Association.

Matthew Shardlow. 2013. 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, Sofia, Bulgaria. Association for Computational Linguistics.

Matthew Shardlow. 2014. Out in the Open: Finding and Categorising Errors in the Lexical Simplification Pipeline. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1583–1590, Reykjavik, Iceland. European Language Resources Association (ELRA).

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.

Seid Muhie Yimam, Sanja Štajner, Martin Riedl, and Chris Biemann. 2017. 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, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. 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, New Orleans, Louisiana. Association for Computational Linguistics.

Xingxing Zhang and Mirella Lapata. 2017. Sentence Simplification with Deep Reinforcement Learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark. Association for Computational Linguistics.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A Monolingual Tree-based Translation Model for Sentence Simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361, Beijing, China. Coling 2010 Organizing Committee.