Lexical Simplification with Neural Ranking

Gustavo Henrique Paetzold and **Lucia Specia**
Department of Computer Science
University of Sheffield, UK
{g.h.paetzold,l.specia}@sheffield.ac.uk

Abstract

We present a new Lexical Simplification approach that exploits Neural Networks to learn substitutions from the Newsela corpus - a large set of professionally produced simplifications. We extract candidate substitutions by combining the Newsela corpus with a retrofitted context-aware word embeddings model and rank them using a new neural regression model that learns rankings from annotated data. This strategy leads to the highest Accuracy, Precision and F1 scores to date in standard datasets for the task.

1 Introduction

In Lexical Simplification (LS), words and expressions that challenge a target audience are replaced with simpler alternatives. Early lexical simplifiers (Devlin and Tait, 1998; Carroll et al., 1998) combine WordNet (Fellbaum, 1998) and frequency information such as Kucera-Francis coefficients (Rudell, 1993). Modern simplifiers are more sophisticated, but most of them still adhere to the following pipeline: Complex Word Identification (CWI) to select words to simplify; Substitution Generation (SG) to produce candidate substitutions for each complex word; Substitution Selection (SS) to filter candidates that do not fit the context of the complex word; and Substitution Ranking (SR) to rank them according to their simplicity.

The most effective LS approaches exploit Machine Learning techniques. In CWI, ensembles that use large corpora and thesauri dominate the top 10 systems in the CWI task of SemEval 2016 (Paetzold and Specia, 2016d). In SG, Horn et al. (2014) extract candidates from a parallel Wikipedia and Simple Wikipedia corpus, yielding major improvements over previous approaches (Devlin, 1999; Biran et al., 2011). Glavaš and Štajner (2015) and Paetzold and Specia (2016f) employ word embedding models to generate candidates, leading to even better results.

In SR, the state-of-the-art performance is achieved by employing supervised approaches: SVMRank (Horn et al., 2014) and Boundary Ranking (Paetzold and Specia, 2015). Supervised approaches have the caveat of requiring annotated data, but as a consequence they can adapt to the needs of a specific target audience.

Recently, (Xu et al., 2015) introduced the Newsela corpus, a new resource composed of thousands of news articles simplified by professionals. Their analysis reveals the potential use of this corpus in simplification, but thus far no simplifiers exist that exploit this resource. The scale of this corpus and the fact that it was created by professionals opens new avenues for research, including using Neural Network approaches, which have proved promising for many related problems.

Neural Networks for supervised ranking have performed well in Information Retrieval (Burges et al., 2005), Medical Risk Evaluation (Caruana et al., 1995) and Summarization (Cao et al., 2015), among other tasks, which suggests that they could be an interesting approach to SR. In the context of LS, existing work has only exploited word embeddings as features for SG, SS and SR.

In this paper, we introduce an LS approach that uses the Newsela corpus for SG and employs a new regression model for Neural Ranking in SR that addresses the task in three steps: Regression, Ordering and Confidence Check.

2 Hybrid Substitution Generation

Our approach combines candidate substitutions from two sources: the Newsela corpus and retrofitted context-aware word embedding models.

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 34–40,
Valencia, Spain, April 3-7, 2017.

2.1 SG via Parallel Data

The Newsela corpus[1] (version 2016-01-29.1) contains 1,911 news articles in their original form, as well as up to 5 versions simplified by trained professionals to different reading levels. It has a total of 10,787 documents, each with a unique article identifier and a version indicator between 0 and 5, where 0 refers to the article's original form, and 5 to its simplest version.

To employ the Newsela corpus in SG, we first produce sentence alignments for all pairs of versions of a given article. To do so, we use paragraph and sentence alignment algorithms from (Paetzold and Specia, 2016g). They align paragraphs with sentences that have high TF-IDF similarity, concatenate aligned paragraphs, and finally align concatenated paragraphs at sentence-level using the TF-IDF similarity between them. Using this algorithm, we produce 550,644 sentence alignments.

We then tag sentences using the Stanford Tagger (Toutanvoa and Manning, 2000), produce word alignments using Meteor (Denkowski and Lavie, 2011), and extract candidates using a strategy similar to that of Horn et al. (2014). First we consider all aligned complex-to-simple word pairs as candidates. Then we filter them by discarding pairs which: do not share the same POS tag, have at least one non-content word, have at least one proper noun, or share the same stem. After filtering, we inflect all nouns, verbs, adjectives and adverbs to all possible variants. We then complement the candidate substitutions from the Newsela corpus using the following word embeddings model.

2.2 SG via Context-aware Word Embeddings

Paetzold and Specia (2016f) present a state-of-the-art simplifier that generates candidates from a context-aware word embeddings model trained over a corpus composed of words concatenated with universal POS tags. We take this approach a step further by incorporating another enhancement: lexicon retrofitting.

Faruqui et al. (2015) introduce an algorithm that allows for typical embeddings to be retrofitted over lexicon relations, such as synonymy, hypernymy, etc. To retrofit the context-aware models from (Paetzold and Specia, 2016f), we concatenate the words in WordNet (Fellbaum, 1998) with their universal POS tag, create a dictionary containing mappings between word-tag pairs and their synonyms, then use the algorithm described in (Faruqui et al., 2015).

We train a bag-of-words (CBOW) model (Mikolov et al., 2013b) of 1,300 dimensions with `word2vec` (Mikolov et al., 2013a) using a corpus of over 7 billion words that includes the SubIMDB corpus (Paetzold and Specia, 2016b), UMBC webbase[2], News Crawl[3], SUBTLEX (Brysbaert and New, 2009), Wikipedia and Simple Wikipedia (Kauchak, 2013). We retrofit the model over WordNet's synonym relations only. We choose this model training configuration because it has been shown to perform best for LS in a recent extensive benchmarking (Paetzold, 2016).

For each target word in the Newsela vocabulary we then generate as complementary candidate substitutions the three words in the model with the lowest cosine distances from the target word that have the same POS tag and are not a morphological variant. As demonstrated by Paetzold and Specia (2016a), in SG parallel corpora tend to yield higher Precision, but noticeably lower Recall than embedding models. We add only three candidates in order increase Recall without compromising the high Precision from the Newsela corpus.

3 Unsupervised Substitution Selection

We pair our generator with the Unsupervised Boundary Ranking SS approach from (Paetzold and Specia, 2016f). They learn a supervised ranking model over data gathered in unsupervised fashion. Candidates are ranked according to how well they fit the context of the target word, and a percentage of the worst ranking candidates is discarded.

For training, the approach requires a set of complex words in context along with candidate substitutions for it. To produce this data, we generate candidates for the complex words in all 929 simplification instances of the BenchLS dataset (Paetzold and Specia, 2016a) using our SG approach. The selector assigns label 1 to the complex words and 0 to all candidates, then trains the model over this data. During SS, we discard 50% of candidates with the worst rankings. We chose this proportion through experimentation. As features, we use the same described in (Paetzold and Specia, 2016f).

[1] https://newsela.com/data

[2] http://ebiquity.umbc.edu/resource/html/id/351

[3] http://www.statmt.org/wmt11/translation-task.html

4 Neural Substitution Ranking

Our approach performs three steps: Regression, Ordering and Confidence Check.

4.1 Regression

In this step, we employ a multi-layer perceptron to determine the ranking between candidate substitutions. The network (Figure 1) takes as input a set of features from two candidates, and produces a single value that represents how much simpler candidate 1 is than candidate 2. If the value is negative, then candidate 1 is simpler than 2, if it is positive, candidate 2 is simpler than 1.

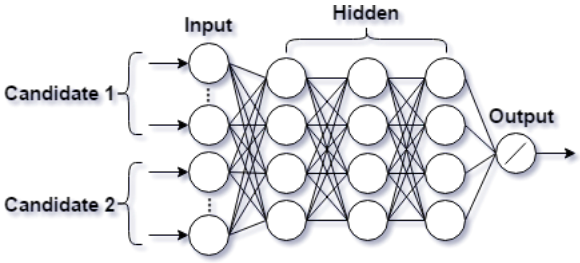

Figure 1: Architecture of neural ranker

Our network has three hidden layers with eight nodes each. For training we use the LexMTurk dataset (Horn et al., 2014), which contains 500 instances composed of a sentence, a target complex word and candidate substitutions ranked by simplicity. Let c_1 and c_2 be a pair of candidates from an instance, r_1 and r_2 their simplicity ranks, and $\Phi(c_i)$ a function that maps a candidate c_i to a set of feature values. For each possible pair in each instance of the LexMTurk dataset we create two training instances: one with input $[\Phi(c_1), \Phi(c_2)]$ and reference output $r_1 - r_2$, and one with input $[\Phi(c_2), \Phi(c_1)]$ and reference output $r_2 - r_1$. We train our model for 500 epochs. We use the same n-gram probability features from SubIMDB used by (Paetzold and Specia, 2015). Hidden layers use the $tanh$ activation function, and the output node uses a linear function with Mean Average Error.

4.2 Ordering

Once the model is trained, we rank candidates by simplicity. Let $M(c_i, c_j)$ be the value estimated by our model for a pair of candidates c_i and c_j of a generated set C. During the ordering, we calculate the final score $R(c_i)$ of all candidates c_i (Eq. 1).

$$R(c_i) = \sum_{c_j \neq c_i \in C} M(c_i, c_j) \quad (1)$$

Then, we simply rank all candidates based on R: the lower the score, the simpler a candidate is.

4.3 Confidence Check

Once candidates are ranked, in order to increase the reliability of our simplifier, instead of replacing the target complex word with the simplest candidate, we first compare the use of this candidate against the original word in context, which can be seen as a Confidence Check.

The target t is only replaced by the simplest candidate c if the language model probability of the trigram $S_{j-2}^{j-1}\, t$, in which S_{j-2}^{j-1} is the bigram of words preceding t in position j of sentence S, is smaller than that of trigram $S_{j-2}^{j-1}\, c$. This type of approach has been proved a reliable alternative to simply adding the target complex word to the candidate pool during ranking (Glavaš and Štajner, 2015).

To calculate probabilities, we train a 5-gram language model over SubIMDB, since its word and n-gram frequencies have been shown to correlate with simplicity better than those from other larger corpora (Paetzold and Specia, 2016b). We henceforth refer to our LS approach (SG+SS+SR) as **NNLS**.

5 Substitution Generation Evaluation

Here we assess the performance of our SG approach in isolation (NNLS/SG), and when paired with our SS strategy (NNLS/SG+SS), as described in Sections 2 and 3. We compare them to the generators of all approaches featured in the benchmarks of Paetzold and Specia (2016a): Devlin (Devlin and Tait, 1998), Biran (Biran et al., 2011), Yamamoto (Kajiwara et al., 2013), Horn (Horn et al., 2014), Glavas (Glavaš and Štajner, 2015) and Paetzold (Paetzold and Specia, 2015; Paetzold and Specia, 2016f). These SG strategies extract candidates from WordNet, Wikipedia and Simple Wikipedia articles, Merriam dictionary, sentence-aligned Wikipedia and Simple Wikipedia articles, typical word embeddings and context-aware word embeddings, respectively. They are all available in the LEXenstein framework (Paetzold and Specia, 2015).

We use two common evaluation datasets for LS: BenchLS (Paetzold and Specia, 2016a), which contains 929 instances and is annotated by English speakers from the U.S, and NNSEval (Paetzold and Specia, 2016f), which contains 239 instances

and is annotated by non-native English speakers. Each instance is composed of a sentence, a target complex word, and a set of gold candidates ranked by simplicity. We use the same metrics featured in (Paetzold and Specia, 2016a), which are the well known Precision, Recall and F1. Notice that, since these datasets already provide target words deemed complex by human annotators, we do not address CWI in our evaluations.

The results in Table 1 reveal that our SG approach outperforms all others in Precision and F1 by a considerable margin, and that our SS approach leads to noticeable increases in Precision at almost no cost in Recall.

	BenchLS			**NNSeval**		
	P	**R**	**F1**	**P**	**R**	**F1**
Devlin	0.133	0.153	0.143	0.092	0.093	0.092
Biran	0.130	0.144	0.136	0.084	0.079	0.081
Yamamoto	0.032	0.087	0.047	0.026	0.061	0.037
Horn	0.235	0.131	0.168	0.134	0.088	0.106
Glavas	0.142	0.191	0.163	0.105	0.141	0.121
Paetzold	0.180	**0.252**	0.210	0.118	**0.161**	0.136
NNLS/SG	0.270	0.209	0.236	0.186	0.136	0.157
NNLS/SG+SS	**0.337**	0.206	**0.256**	**0.231**	0.135	**0.171**

Table 1: SG benchmarking results

6 Substitution Ranking Evaluation

We also compare our Neural Ranking SR approach (NNLS/SR) to the rankers of all aforementioned lexical simplifiers. The Devlin, Biran, Yamamoto, Horn, Glavas and Paetzold rankers exploit Kucera-Francis coefficients (Rudell, 1993), hand-crafted complexity metrics, a supervised SVM ranker, rank averaging and Boundary Ranking, respectively. In this experiment we disregard the step of Confidence Check, since we aim to analyse the performance of our ranking strategy alone.

The datasets used are those introduced for the English Lexical Simplification task of SemEval 2012 (Specia et al., 2012), to which dozens of systems were submitted. The training and test sets are composed of 300 and 1,710 instances, respectively. Each instance is composed of a sentence, a target complex word, and a series of candidate substitutions ranked by simplicity. We use TRank, the official metric of the SemEval 2012 task, which measures the proportion of instances for which the candidate with the highest gold-rank was ranked first, as well Pearson (p) correlation. While TRank best captures the reliability of rankers in practice, Pearson correlation shows how well the rankers capture simplicity in general.

Table 2 reveals that, much like our SG approach, our Neural Ranker performs well in isolation, offering the highest scores among all strategies available.

	TRank	p
Devlin	0.596	0.614
Biran	0.513	0.505
Yamamoto	0.604	0.649
Horn	0.639	0.673
Glavas	0.632	0.644
Paetzold	0.653	**0.677**
NNLS/SR	**0.658**	**0.677**

Table 2: SR benchmarking results

7 Full Pipeline Evaluation

We then evaluate our approach in two settings: with (NNLS) and without (NNLS-C), the Confidence Check (Section 4.3). The evaluation datasets used are the same described in Section 5, and the metrics are:

- **Accuracy:** The proportion of instances in which the target word was replaced by a gold candidate.
- **Precision:** The proportion of instances in which the target word was either replaced by a gold candidate or not replaced at all.

	BenchLS		**NNSeval**	
	P	**A**	**P**	**A**
Devlin	0.309	0.307	0.335	0.117
Biran	0.124	0.123	0.121	0.121
Yamamoto	0.044	0.041	0.444	0.025
Horn	0.546	0.341	0.364	0.172
Glavas	0.480	0.252	0.456	0.197
Paetzold	0.423	0.423	0.297	0.297
NNLS	**0.642**	0.434	**0.544**	0.335
NNLS-C	0.543	**0.538**	0.397	**0.393**

Table 4: Full pipeline evaluation results

Notice that, unlike in SG, Recall and F1 are not applicable in this form of evaluation. Table 4 reveals that, without the confidence check, our approach yields an average increase of 10.5% in Accuracy over the former state-of-the-art simplifier. With the confidence check, it yields the highest Precision while retaining the highest Accuracy.

		2A	2B	3A	3B	4	5	1
SE	Devlin	**0 (0%)**	689 (74%)	86 (36%)	34 (14%)	60 (50%)	17 (14%)	43 (36%)
SE	Horn	**0 (0%)**	689 (74%)	76 (32%)	43 (18%)	74 (61%)	**15 (12%)**	32 (26%)
SE	Glavas	**0 (0%)**	689 (74%)	70 (29%)	23 (10%)	81 (55%)	20 (14%)	46 (31%)
SE	Paetzold	**0 (0%)**	689 (74%)	59 (25%)	**21 (9%)**	68 (42%)	28 (18%)	64 (40%)
SE	NNLS	**0 (0%)**	689 (74%)	**40 (17%)**	30 (12%)	**34 (20%)**	45 (26%)	**91 (54%)**
PV	Devlin	84 (9%)	**232 (25%)**	146 (61%)	22 (9%)	35 (49%)	**8 (11%)**	29 (40%)
PV	Horn	84 (9%)	**232 (25%)**	123 (51%)	30 (12%)	50 (57%)	13 (15%)	24 (28%)
PV	Glavas	84 (9%)	**232 (25%)**	127 (53%)	12 (5%)	46 (46%)	17 (17%)	38 (38%)
PV	Paetzold	84 (9%)	**232 (25%)**	126 (52%)	**9 (4%)**	39 (37%)	14 (13%)	52 (50%)
PV	NNLS	84 (9%)	**232 (25%)**	**110 (46%)**	17 (7%)	**14 (12%)**	26 (23%)	**73 (65%)**

Table 3: Error categorisation results

8 Error Analysis

In this Section we analyse NNLS to understand the sources of its errors. For that, we use PLUMBErr (Paetzold and Specia, 2016c; Shardlow, 2014), a method that assesses all steps taken by LS systems and identifies five types of errors:

- **1:** No error during simplification.
- **2A:** Complex word classified as simple.
- **2B:** Simple word classified as complex.
- **3A:** No candidate substitutions produced.
- **3B:** No simpler candidates produced.
- **4:** Replacement compromises the sentence's grammaticality or meaning.
- **5:** Replacement does not simplify the word.

Errors of type 2 are made during CWI, 3 during SG/SS, and 4 and 5 during SR. We pair ours, Devlin's, Horn's, Glavas' and Paetzold's simplifiers with two CWI approaches: one that simplifies everything (SE), and the Performance-Oriented Soft Voting approach (PV), which won the CWI task of SemEval 2016 (Paetzold and Specia, 2016e).

Table 3 shows the count and proportion (in brackets) of instances in BenchLS in which each error was made. It shows that our approach correctly simplifies the largest number of problems, while making the fewest errors of type 3A and 4. However, it can be noticed that NNLS makes many errors of type 5. By analysing the output produced after each step, we found that this is caused by the inherently high Precision of our approach: by producing a smaller number of spurious candidates, our simplifier reduces the occurrences of ungrammatical and/or incoherent substitutions, but also disregards many candidates that are simpler than the target complex word. Nonetheless, this noticeably increases the number of correct simplifications made.

9 Conclusions

We introduced an LS approach that extracts candidate substitutions from the Newsela corpus and retrofitted context-aware word embedding models, selects them with Unsupervised Boundary Ranking, and ranks them using a new Neural Ranking strategy.

We found that: (i) our generator achieves the highest Precision and F1 scores to date, (ii) our Neural Ranking strategy leads to the top scores on the English Lexical Simplification task of SemEval 2012, (iii) and their combination offers the highest Precision and Accuracy scores in two standard evaluation datasets. An error analysis reveals that our LS approach makes considerably fewer grammaticality/meaning errors than former state-of-the-art simplifiers.

In future work, we aim to investigate new architectures for our Neural Ranking model, as well as to test our approach in other NLP tasks. An implementation of our Substitution Generation, Selection and Ranking approaches can be found in the LEXenstein framework[4].

Acknowledgements

This work has been supported by the European Commission project SIMPATICO (H2020-EURO-6-2015, grant number 692819).

[4]http://ghpaetzold.github.io/LEXenstein

References

Or Biran, Samuel Brody, and Noemie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 496–501, Portland, Oregon, USA, June. Association for Computational Linguistics.

Marc Brysbaert and Boris New. 2009. Moving beyond kučera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american english. *Behavior research methods*, 41:977–990.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine learning*, pages 89–96. ACM.

Ziqiang Cao, Furu Wei, Li Dong, Sujian Li, and Ming Zhou. 2015. Ranking with recursive neural networks and its application to multi-document summarization. In *Proceedings of the 2015 AAAI*, pages 2153–2159, Austin, USA.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. In *Proceedings of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10, Madison, USA.

Rich Caruana, Shumeet Baluja, and Tom Mitchell. 1995. Using the future to "sort out" the present: Rankprop and multitask learning for medical risk evaluation. In *Proceedings of the 8th International Conference on Neural Information Processing Systems*, NIPS'95, pages 959–965, Denver, Colorado. MIT Press.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland, July. Association for Computational Linguistics.

Siobhan Devlin and John Tait. 1998. The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic Databases*, pages 161–173.

Siobhan Devlin. 1999. *Simplifying Natural Language for Aphasic Readers*. Ph.D. thesis, University of Sunderland.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado, May–June. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 63–68, Beijing, China, July. Association for Computational Linguistics.

Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a lexical simplifier using wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 458–463, Baltimore, Maryland, June. Association for Computational Linguistics.

Tomoyuki Kajiwara, Hiroshi Matsumoto, and Kazuhide Yamamoto. 2013. Selecting proper lexical paraphrase for children. In *Proceedings of the 25th Conference on Computational Linguistics and Speech Processing*, pages 59–73, Kaohsiung, Taiwan.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1537–1546, Sofia, Bulgaria, August. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26:3111–3119.

Gustavo Henrique Paetzold and Lucia Specia. 2015. Lexenstein: A framework for lexical simplification. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 85–90, Beijing, China, July. Association for Computational Linguistics and The Asian Federation of Natural Language Processing.

Gustavo Henrique Paetzold and Lucia Specia. 2016a. Benchmarking lexical simplification systems. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portoroz, Slovenia. European Language Resources Association (ELRA).

Gustavo Henrique Paetzold and Lucia Specia. 2016b. Collecting and exploring everyday language for predicting psycholinguistic properties of words. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1669–1679, Osaka, Japan, December.

Gustavo Henrique Paetzold and Lucia Specia. 2016c. Plumberr: An automatic error identification framework for lexical simplification. In *Proceedings of the 1st Workshop on Quality Assessment for Text Simplification*, pages 7–15, Portoroz, Slovenia. European Language Resources Association (ELRA).

Gustavo Henrique Paetzold and Lucia Specia. 2016d. Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569, San Diego, California, June. Association for Computational Linguistics.

Gustavo Henrique Paetzold and Lucia Specia. 2016e. Sv000gg at semeval-2016 task 11: Heavy gauge complex word identification with system voting. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 969–974, San Diego, California, June. Association for Computational Linguistics.

Gustavo Henrique Paetzold and Lucia Specia. 2016f. Unsupervised lexical simplification for non-native speakers. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3761–3767. AAAI Press.

Gustavo Henrique Paetzold and Lucia Specia. 2016g. Vicinity-driven paragraph and sentence alignment for comparable corpora. *arXiv preprint arXiv:1612.04113*.

Gustavo Henrique Paetzold. 2016. *Lexical Simplification for Non-Native English Speakers*. Ph.D. thesis, University of Sheffield.

Allan Peter Rudell. 1993. Frequency of word usage and perceived word difficulty: Ratings of kucera and francis words. *Behavior Research Methods*, pages 455–463.

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)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. Semeval-2012 task 1: English 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 347–355, Montréal, Canada. Association for Computational Linguistics.

Kristina Toutanvoa and Christopher Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 63–70, Hong Kong, China, October. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.