Simple PPDB: A Paraphrase Database for Simplification

Anonymous ACL submission

Abstract

We release the Simple Paraphrase Database, a subset of of the Paraphrase Database (PPDB) adapted for the task of text simplification. We train a supervised model to associate simplification scores with each phrase pair, producing rankings competitive with state-of-the-art lexical simplification models. Our new simplification database contains 4.4 million paraphrase rules, making it the largest available resource for lexical simplification.

1 Motivation

Language is complex, and the process of reading and understanding language is difficult for many groups of people. The goal of text simplification is to rewrite text in order to make it easier to understand, for example, by children (De Belder and Moens, 2010), language learners (Petersen and Ostendorf, 2007), people with disabilities (Rello et al., 2013; Evans et al., 2014), and even by machines (Siddharthan et al., 2004). Automatic text simplification (Napoles and Dredze, 2010; Wubben et al., 2012; Anonymous, provisionally accepted) has the potential to dramatically increase access to information by making written documents available at all reading levels.

Full text simplification involves many steps, including grammatical restructuring and summarization (Feng, 2008). One of the most basic subtasks is *lexical simplification* (Specia et al., 2012)– replacing complicated words and phrases with simpler paraphrases. While there is active research in the area of lexical simplification (Coster and Kauchak, 2011a; Glavaš and Štajner, 2015; Paetzold, 2015), existing models have been by-and-large limited to single words. Often, however, it is preferable, or even necessary to paraphrase a single complex word with multiple simpler words, or to paraphrase multiple words with a single word. For example, it is difficult to imagine a simple, single-word paraphrase of *hypertension*, but the three-word phrase *high blood pressure* is a very good simplification (Table 1). Such phrasal simplifications are overlooked by current lexical simplification models, and thus are often unavailable to the end-to-end text simplification systems that require them.

medical practitioner	→	doctor
internationally renowned	→	famous
hypertension	→	high blood pressure
prevalent	→	really common
significant quantity	→	a lot
impact negatively	→	be bad

Table 1: In lexical simplification, it is often necessary to replace single words with phrases or phrases with single words. The above are examples of such lexical simplifications captured by the Simple PPDB resource.

Recent research in data-driven paraphrasing has produced enormous resources containing millions of meaning-equivalent words and phrases (Ganitkevitch et al., 2013). Such resources capture a wide range of language variation, including the types of lexical and phrasal simplifications just described. In this work, we apply state-of-the-art machine learning models for lexical simplification in order to identify phrase pairs from the Paraphrase Database (PPDB) which are especially applicable to the task of text simplification. We introduce Simple PPDB, a subset of the Paraphrase Database containing 4.4 million simplifying paraphrase rules. The large scale of Simple PPDB will support research into increasingly advanced methods for text simplification.

2 Identifying Simplification Rules

2.1 Paraphrase Rules

The Paraphrase Database (PPDB) is currently the largest available collection of paraphrases. Each paraphrase rule in the database has an automatically-assigned quality score between 1 and 5 (Pavlick et al., 2015). In this work, we use the PPDB-TLDR dataset, which contains 14 million high-scoring lexical and phrasal paraphrases, and is intended to give a generally good tradeoff between precision and recall. To preprocess the data, we lemmatize all of the phrases, and remove rules which differ only by morphology, punctuation, or stop words, or which involve phrases longer than 3 words. The resulting list contains 7.5 million paraphrase rules covering 625K unique lemmatized words and phrases.

2.2 Lexical Simplification Model

Data. We train a logistic regression model[1] to predict whether or not the application of a paraphrase rule will result in simpler output. Our training data comes from Pavlick and Nenkova (2015), who released 8,286 paraphrase pairs manually labeled for comparative complexity by 7 annotators each. We remove pairs which differ only by morphology, and pairs which have been judged as having no difference in complexity. Our final training dataset consists of 6,118 word and phrase pairs with binary labels, where a label of '1' indicates that the first word in the pair is simpler than the second. Every pair appears twice, once with each ordering, so the label distribution is exactly 50/50.

Features. We use a variety of features that have been shown in prior work to give good signal about phrases' relative complexity. For each phrase pair $\langle e_1, e_2 \rangle$, for each feature f, we include $f(e_1)$, $f(e_2)$ and $f(e_1) - f(e_2)$.[2] The features we include are as follows: phrase length in words and in characters, frequency according to the Google NGram corpus (Brants and Franz, 2006), number of syllables, the relative frequency of usage in Simple Wikipedia compared to normal Wikipedia (Pavlick and Nenkova, 2015), character unigrams and bigrams, POS tags, and averaged word embeddings for the words in the phrase (Mikolov et al., 2013).

[1] http://scikit-learn.org/

[2] We do not compute the difference $f(e_1) - f(e_2)$ for sparse features, i.e. character ngrams and POS tags.

Random	50.0%
Number of Syllables	68.3%
Length in Characters	70.4%
Google Ngram Frequency	70.8%
Simple/Regular Wiki. Ratio	72.5%
Supervised Model, W2V only	76.6%
Supervised Model, Full	**82.7%**

Table 2: Accuracy on 10-fold cross-validation. Folds are constructed so that train and test vocabularies are disjoint.

Performance. Table 2 shows the performance of the model on cross-validation, compared to several baselines. The full model achieves 83% accuracy, 10 points higher than the strongest baseline (the Pavlick and Nenkova (2015) score), which is based on word frequencies in Simple Wikipedia.

2.3 Simple PPDB

We run the trained model described above over all 7.5 million paraphrase rules. From the predictions, we construct Simple PPDB: a list of 4.4 million simplifying paraphrase rules. A rule in Simple PPDB is represented as a triple, consisting of a syntactic category, and input phrase, and a simplified output phrase. Each rule is associated with both a paraphrase quality score from 1 to 5 (taken from PPDB 2.0), and simplification confidence score from 0.5 to 1.0 (our classifier's confidence in the prediction). The remainder of this paper evaluates the quality of the simplification ranking. For an evaluation of the paraphrase quality ranking, see Pavlick et al. (2015). Table 3 shows examples of some of the top ranked simplifications according to Simple PPDB for several input phrases.

3 Evaluation

To evaluate Simple PPDB, we apply it in a setting intended to emulate the way it is likely to be used in practice. We use the Newsela Simplification Dataset (Xu et al., 2015), a corpus of manually simplified news articles. This corpus is currently the cleanest available simplification dataset and is likely to be used to train and/or evaluate the simplification systems that we envision benefitting most from Simple PPDB.

We draw a sample of 100 unique word types ("targets") from the corpus for which Simple PPDB has at least one candidate simplification. For each target, we take Simple PPDB's full list of

keenly	omit	employment opportunity	actively participate
very	forgotten something	job	take part
most strongly	leave out	the unemployed	join the
so strongly	delete it	employee	be a part
so sorely	skip	their use	will play
so urgently	be overlook	the hiring	participate in

Table 3: Examples of top-ranked simplifications proposed by Simple PPDB for several input words. Often, the best simplification for a single word is a multiword phrase, or vis-versa. These many-to-one mappings are overlooked when systems use only length or frequency as a proxy for simplicity.

simplification rules which are of acceptable quality (according to the PPDB 2.0 paraphrase score) and which match the syntactic category of the target. On average, Simple PPDB proposes 8.5 such candidate simplifications per target.

Comparison to existing methods. We also try several existing methods for generating ranked lists of candidate paraphrases for a given target. We test three previously-proposed methods for generating lists of paraphrase candidates: the **WordNetGenerator**, which pulls synonyms from WordNet (Devlin and Tait, 1998; Carroll et al., 1999), the **KauchakGenerator**, which generates candidates based on automatic alignments between Simple Wikipedia and normal Wikipedia (Coster and Kauchak, 2011a), and the **GlavasGenerator**, which generates candidates from nearby phrases in vector space (Glavaš and Štajner, 2015) (we use the pre-trained Word2Vec VSM (Mikolov et al., 2013)).

For each generated list, we use Horn et al. (2014)'s supervised SVM Rank approach. We reimplement the main features of their model: namely, word frequencies according to the Google NGrams corpus (Brants and Franz, 2006) and the Simple Wikipedia corpus, and the alignment probabilities according to automatic word alignments between Wikipedia and Simple Wikipedia sentences (Coster and Kauchak, 2011b). We omit the language modeling features since our evaluation does not consider the context in which the substitution is to be applied.

All of the above methods (the three generation methods and the ranker) are implemented as part of the LEXenstein toolkit (Paetzold and Specia, 2015). We use the LEXenstein implementations for the results reported here, using off-the-shelf configurations and treating each method as a black box.

Setup. We use each of the generate-and-rank methods to produce a ranked list of simplification candidates for each of the 100 targets drawn from the Newsela corpus. When a generation method fails to produce any candidates for a given target, we simply ignore that target for that particular method. This is to avoid giving Simple PPDB an unfair advantage, since, by construction, PPDB will have full coverage of our list of 100 targets. In total, Simple PPDB is evaluated over all 100 targets, the GlavasGenerator over 95, the WordNetGenerator over 82, and the KauchakGenerator over 48. Since the GlavasGenerator is capable of producing an arbitrary number of candidates for each target, we limit the length of each of its candidate lists to be equal to the number of candidates produced by Simple PPDB for that same target.

Human judgments. We collect human judgements for each of the sampled rules on Amazon Mechanical Turk, using the same HIT design as in Pavlick and Nenkova (2015). That is, we ask 7 workers to view each pair and indicate which of the two phrases is simpler, or to indicate that there is no difference. We take the majority label to be the true label for each rule. Workers show moderate agreement on the 3-way task (κ = 0.4 $\pm$ 0.03), with 14% of pairs receiving unanimous agreement and 37% receiving the same label from 6 out of 7 annotators. We note that the κ metric is likely a lower bound, as it punishes low agreement on pairs for which there is little difference in complexity, and thus the "correct" answer is not clear (e.g. for the pair $\langle$*matter, subject*$\rangle$, 3 annotators say that *matter* is simpler, 2 say that *subject* is simpler, and 2 say there is no difference).

Results. Table 4 compares the different methods in terms of how well they rank simplifying rules above non-simplifying rules. Simple PPDB's ranking of the relative simplicity achieves an averaged precision of 0.78 (0.91 P@1), compared

	Avg. Prec.	P@1
Glavas+SVR	0.19	0.13
Wordnet+SVR	0.53	0.50
Kauchak+SVR	0.70	0.69
Simple PPDB	**0.78**	**0.91**

Table 4: Precision of relative simplification rankings of three existing lexical simplification methods compared to the Simple PPDB resource in terms of Average Precision and P@1 (both range from 0 to 1 and higher is better). All of the existing methods were evaluated using the implementations as provided in the LEXenstein toolkit.

to 0.70 (0.69 P@1) achieved by the Horn et al. (2014) system– i.e. the KauchakGenerator+SVM Ranker. We hypothesize that the performance difference between these two ranking systems is likely due to a combination of the additional features applied in Simple PPDB's model (e.g. word embeddings) and the difference in training data (Simple PPDB's model was trained on 6K paraphrase pairs with binary labels, while the Horn et al. (2014) model was trained on 500 words, each with a ranked list of paraphrases). Table 5 provides examples of the top-ranked simplification candidates proposed by each of the methods described.

	alarm
Glavas	**enrage**, perturb, stun
WordNet	horrify, dismay, **alert**, appall, appal
Kauchak	**pure**, **worry**
PPDB	**trouble**, **alert**, **concern**

	genuine
Glavas	credible, **sort**, feign, **phoney**, good naturedness, **sincere**, **sincerely**, insincere, bonafide
WordNet	**real**, **actual**, unfeigned, literal, echt, **true**
Kauchak	thermal
PPDB	**really**, **true**, quite, **honest**, **real**, **and**, legitimate

Table 5: Examples of candidate simplifications proposed by Simple PPDB and by three other generate-and-rank methods. Bold words were rated by humans to be simpler than the target word.

In addition, Simple PPDB offers the largest coverage (Table 6). It has a total vocabulary of 624K unique words and phrases, and provides the largest number of potential simplifications for each target– for the 100 targets drawn from the Newsela corpus, PPDB provided an average of 8.5 candidates per target. The next best generator, the WordNet-based system, produces only 6.7 candidates per target on average, and has a total vocabulary of only 155K words.

	Avg. PPs per Input	Total Vocab.
Glavas+SVR	∞	∞
Kauchak+SVR	4.4	127K
Wordnet+SVR	6.7	155K
Simple PPDB	8.5	624K

Table 6: Overall coverage of three existing lexical simplification methods compared to the Simple PPDB resource.

4 Conclusion

We have described Simple PPDB, a subset of the Paraphrase Database adapted for the task of text simplification. Simple PPDB is built by applying state-of-the-art machine learning models for lexical simplification to the largest available resource of lexical and phrasal paraphrases, resulting in a web-scale resource capable of supporting research in data-driven methods for text simplification. We have shown that Simple PPDB offers substantially increased coverage of both words and multiword phrases, while maintaining high quality compared to existing methods for lexical simplification. Simple PPDB is freely available with the publication of this paper.[3]

References

Anonymous. provisionally accepted. Optimizing statistical machine translation for simplification. *Transactions of the Association for Computational Linguistics*.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram Version 1. *Linguistic Data Consortium, Philadelphia*.

John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. Simplifying text for language-impaired readers. In *Proceedings of EACL*, volume 99, pages 269–270.

Will Coster and David Kauchak. 2011a. Learning to simplify sentences using Wikipedia. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 1–9, Portland, Oregon, June. Association for Computational Linguistics.

William Coster and David Kauchak. 2011b. Simple English Wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669, Portland, Oregon, USA, June. Association for Computational Linguistics.

[3] `http://...`

Jan De Belder and Marie-Francine Moens. 2010. Text simplification for children. In *Proceedings of the SIGIR workshop on accessible search systems*, pages 19–26. ACM.

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.

Richard Evans, Constantin Orasan, and Iustin Dornescu. 2014. An evaluation of syntactic simplification rules for people with autism. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 131–140.

Lijun Feng. 2008. Text simplification: A survey. *The City University of New York, Tech. Rep*.

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, June. Association for Computational Linguistics.

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.

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

Courtney Napoles and Mark Dredze. 2010. Learning Simple Wikipedia: A cogitation in ascertaining abecedarian language. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids*, pages 42–50, Los Angeles, CA, USA, June. Association for Computational Linguistics.

Gustavo 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 Paetzold. 2015. Reliable lexical simplification for non-native speakers. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 9–16, Denver, Colorado, June. Association for Computational Linguistics.

Ellie Pavlick and Ani Nenkova. 2015. Inducing lexical style properties for paraphrase and genre differentiation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 218–224, Denver, Colorado, May–June. Association for Computational Linguistics.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. 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 425–430, Beijing, China, July. Association for Computational Linguistics.

Sarah E. Petersen and Mari Ostendorf. 2007. Text simplification for language learners: a corpus analysis. In *SLaTE*, pages 69–72. Citeseer.

Luz Rello, Ricardo A. Baeza-Yates, and Horacio Saggion. 2013. The impact of lexical simplification by verbal paraphrases for people with and without dyslexia. In *CICLing (2)*, pages 501–512.

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

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, 7-8 June. Association for Computational Linguistics.

Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 1015–1024. 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.