teamPN at TSAR-2022 Shared Task: Lexical Simplification Using Multi-Level and Modular Approach

Nikita Katyal
nikita18katyal@gmail.com

Pawan Kumar Rajpoot
pawan.rajpoot2411@gmail.com

Abstract

Lexical Simplification is the process of reducing the lexical complexity of a text by replacing difficult words with easier to read (or understand) expressions while preserving the original information and meaning. This paper explains the work done by our team "teamPN" for English track of TSAR 2022 Shared Task of Lexical Simplification. We created a modular pipeline which combines transformers based models with traditional NLP methods like paraphrasing and verb sense disambiguation. We created a multi level and modular pipeline where the target text is treated according to its semantics (Part of Speech Tag). Pipeline is multi level as we utilize multiple source models to find potential candidates for replacement. It is modular as we can switch the source models and their weighting in the final re-ranking.

1 Introduction

As per TSAR-2022 Workshop Shared Task the problem definition is: "Given a sentence containing a complex word, systems should return an ordered list of simpler valid substitutes for the complex word in its original context. The list of simpler words (up to a maximum of 10) returned by the system should be ordered by the confidence the system has in its prediction (best predictions first) and it must not contain ties." One example is shown in Table 1. The English data-set consists of 373 sentences, with 1 complex word per sentence. No training data was provided and the teams were free to create supervised or unsupervised model. We found that majority of the complex words were verbs or nouns (see Table 2). If not noun or verb, we consider the POS to be of "Others" type. This motivated us to build a pipeline where we first disambiguate the words and then find optimal substitutes. Verbs and nouns are generally more ambiguous in the senses which they are used when compared to other Part of Speech tags. We based our whole idea on this assumption and hence treated verbs and nouns with an additional module. Other than verb/noun only module we have 2 modules which we use for all the POS tags. First common to all module uses Distil BERT based word prediction, while the second one uses Paraphrase Database to do a standard lookup for finding potential substitute candidates. Verb only module is based around Verbnet where we do verb sense disambiguation and then as per predicted verb class we collect potential substitute candidates.
Noun only module first grounds the noun entity to a standard knowledge graph. Once entity is grounded we parse the surrounding neighbours from the KG and collect potential substitute candidates.
Once all modules individually run, all potential candidates are combined and re-ranked using Transformer based model.

Sentence	Substitutes
That prompted the military to **deploy** its largest warship, the BRP Gregorio del Pilar.	send, post, use, position, employ, extend, launch

Table 1: Example sentence with complex word (in red) and substitutes (in teal).

Nouns	Verbs	Others
162	145	66

Table 2: POS tags of complex words in TSAR 2022 Shared Task en evaluation data.

2 Approach

We parse the sentence using spacy (Honnibal and Montani, 2017) and run different sets of modules for verb, noun and others respectively. Our modules are explained in in detail as follows. See Algorithm 1 for pseudo code of the pipeline.

Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022), pages 239 - 242
December 8, 2022 ©2022 Association for Computational Linguistics

2.1 Potential Candidate Collection

2.1.1 Verb Sense Disambiguation

Verbnet is a lexicon which is an extension to Levin's original verb classifications (Levin, 1993) in 1993. Semantically similar verbs are placed in same class. We use Verbnet 3.1 (Schuler, 2005) to ground the verb and get possible classes. For class prediction we do not rely on traditional VSD work (Abend et al., 2008; Dligach and Palmer, 2008; Kawahara and Palmer, 2014) as the data which is used in model training is Wall Street Journal historical text data (Loper et al., 2007) which is biased towards fintech domain. For instance the verb "rise" has 6 possible classes in verbnet, but in WSJ data 93 percent of the examples have "rise" related to "calibration" class, as in "Stocks rise from 10 to 12". There have been related research where efficiency of BERT (Devlin et al., 2018) model to capture English syntactic phenomena is studied (Goldberg, 2019), this motivated us to instead do transformer based VSD (see Figure 1) . We first mask the target word and use FitBERT (Havens and Stal, 2019) to rank the top possible words among all possible classes member verbs. As per the work[1] "FitBERT is trained to look at a sentence with a blank, and output an ordered list of every possible word that could fill in that blank, and a score indicating how likely that word is". We choose the verbnet class with maximum representations in top k predicted words. Once the class is fixed we return the class members as potential candidates.

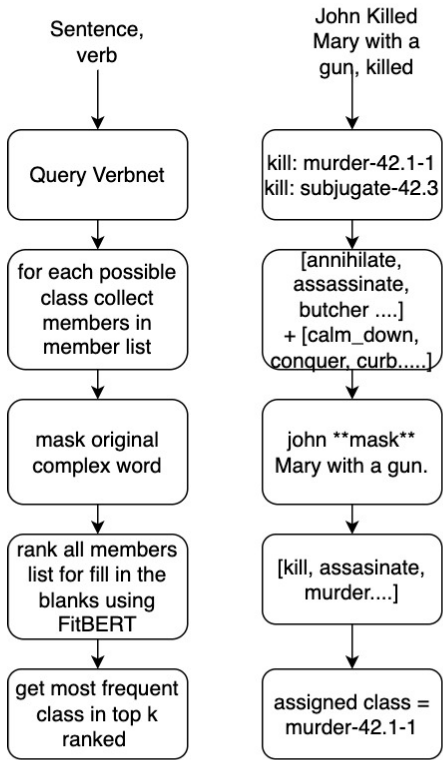

Figure 1: Verb Sense Disambiguation Module. Left part explains overall flow. Right part shows how one example passes through the module.

2.1.2 Paraphrase DataBase

We directly query PPDB (Ganitkevitch et al., 2013) and return the retrieved result list as potential candidates. We use lexical version and small size dictionary of PPDB as it contains the highest quality paraphrases. We use PPDB python library[2] for loading and querying the database.

2.1.3 Distil BERT

DistilBERT (Sanh et al., 2019) is a transformers model which is smaller and faster than BERT, which was pretrained on the same corpus in a self-supervised manner. It is based on Knowledge distillation (teacher student) (Bucila et al., 2006; Hinton et al., 2015) method. Rather than training with a cross-entropy over the hard targets (one-hot encoding of the gold class), knowledge is transfered from the teacher (BERT) to the student (DistilBERT) with a cross-entropy over the soft targets (probabilities of the teacher). We mask the complex word in the context and then use DistilBERT model to predict the words (fill-mask pipeline) then return the result list as the potential candidates. Due to computational resource restrictions we were not able to use complex Transformer models.

2.1.4 Knowledge Graph

We use Multi Modal Knowledge Graph VisualSem (Alberts et al., 2020) to do text entity extraction and grounding to KG for the target complex word. For entity extraction, CLIP textual embedding (Radford et al., 2021) were used as defined in original paper. Retrieval is implemented with k nearest neighbour where the dot-product between the sentence vector and all nodes' gloss matrix for VisualSem graph is calculated. Top-k unique nodes associated to the most relevant glosses are retrieved and if they are same as complex word, the corresponding synonym neighbours are added to the potential candidate list.

[1] https://medium.com/@samhavens/introducing-fitbert-4b047af860fd
[2] https://github.com/erickrf/ppdb

2.2 Aggregation and re-ranking

See Table 3 for usage of modules as per POS tags. Once all potential candidate list is created first we combine all together, then we adjust all the inflections. For inflection correction we use pattern[3] library. We inflect the all candidate words with same tense and quantity (singular/plural) as complex word. Then we again use FitBERT (Havens and Stal, 2019) to rank the combined candidates. For the submissions we used 5 top words.

Algorithm 1 teamPN: Text Simplification

```
Require: m1 = vsdModule
Require: m2 = PPDBModule
Require: m3 = distilBertModule
Require: m4 = kgModule
  for each sentence and complexWord do
    pos = getPos(complexWord)
    if pos == verb then
      candidates = m1 + m2 + m3 + m4
    end if
    if pos == noun then
      candidates = m2 + m3 + m4
    end if
    if pos == Others then
      candidates = m2 + m3
    end if
    candidates = fixInflection(candidates)
    rankCandidates = rerankUsingFitBERT
  end for
```

POS/Module	VSD	PPDB	distil BERT	KG
VERB	Y	Y	Y	N
NOUN	N	Y	Y	Y
Others	N	Y	Y	N

Table 3: Use of Candidate collection modules as per part of Speech of complex word.

3 Results

As per TSAR definition (Štajner et al., 2022) The evaluation metrics to be applied in the TSAR-2022 Shared Task are the following:

MAP@K (Mean Average Precision @ K): K=1,3,5,10. The MAP@K metric is used to check whether the predicted word can be matched (relevant) or not matched (irrelevant) against the set of the gold-standard annotations for evaluation. MAP@K for Lexical Simplification evaluates the following aspects: 1) are the predicted substitutes relevant?, and 2) are the predicted substitutes at the top positions?

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

Accuracy@K@top1: K=1,2,3. The ratio of instances where at least one of the K top predicted candidates matches the most frequently suggested synonym/s in the gold list of annotated candidates.

We stand 12th, on the official results[4] (Saggion et al., 2022) of TSAR-2022 Shared Task. We outperform one of the baseline models TUNER (Štajner et al., 2022). See Table 4 for our scores. We submitted output from 3 different runs, the only difference between the 3 versions was the value for threshold for DistilBERT unmasker module. This threshold corresponds for the minimum confidence cut off for the words predicted. See Table 5 for the threshold values used.

Metric	Run 1	Run 2	Run 3
ACC@1	0.4477	**0.4664**	0.4504
ACC@1@Top1	0.1769	**0.1823**	0.1769
ACC@2@Top1	0.2815	**0.3056**	0.2841
ACC@3@Top1	0.3297	**0.3378**	0.3297
MAP@3	0.2666	**0.2743**	0.2676
MAP@5	0.1874	**0.195**	0.1872
MAP@10	0.0937	**0.0975**	0.0936
Potential@3	0.6621	**0.6729**	0.6648
Potential@5	0.7453	**0.7506**	0.7399
Potential@10	0.7453	**0.7506**	0.7399

Table 4: Our scores for TSAR 2022 Shared Task -EN track

Run 1	Run 2	Run 3
0.02	0.03	0.01

Table 5: DitilBERT Threshold values for 3 runs.

[3] https://github.com/clips/pattern

[4] https://taln.upf.edu/pages/tsar2022-st/#results

4 Conclusion and Future Work

We presented a novel approach where we combine power of transformer based models with traditional NLP. Our work was restricted by computing resources. We would further like to improve on using more modules built out from complex transformers. Also apart from PPDB we did not work with any other synonym dictionaries, adding more open source dictionary modules will bring on more variety. All of our code and documentation is available on Github[5].

Acknowledgements

We would like to acknowledge TSAR organizing committee and EMNLP 2022 for their support and also organizing the Workshop event.

References

Omri Abend, Roi Reichart, and Ari Rappoport. 2008. A supervised algorithm for verb disambiguation into verbnet classes. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, pages 9–16.

Houda Alberts, Teresa Huang, Yash Deshpande, Yibo Liu, Kyunghyun Cho, Clara Vania, and Iacer Calixto. 2020. Visualsem: a high-quality knowledge graph for vision and language. *CoRR*, abs/2008.09150.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *KDD*, pages 535–541. ACM.

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

Dmitriy Dligach and Martha Palmer. 2008. Novel semantic features for verb sense disambiguation. In *Proceedings of ACL-08: HLT, Short Papers*, pages 29–32.

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.

Yoav Goldberg. 2019. Assessing bert's syntactic abilities. *CoRR*, abs/1901.05287.

Sam Havens and Aneta Stal. 2019. Use bert to fill in the blanks.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Daisuke Kawahara and Martha Palmer. 2014. Single classifier approach for verb sense disambiguation based on generalized features. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4210–4213.

Beth Levin. 1993. *English verb classes and alternations: A preliminary investigation*. University of Chicago press.

Edward Loper, Szu-Ting Yi, and Martha Palmer. 2007. Combining lexical resources: mapping between propbank and verbnet. In *Proceedings of the 7th International Workshop on Computational Linguistics, Tilburg, the Netherlands*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020.

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

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Karin Kipper Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania.

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

[5] https://github.com/katyalnikita/TSAR-2022-teamPN