Improving Synonym Recommendation Using Sentence Context

Maria Glenski
Pacific Northwest National Laboratory
Maria.Glenski@pnnl.gov

William Sealy
Georgia Institute of Technology
william.sealy@gatech.edu

Kate Miller
Pacific Northwest National Laboratory
Kate.Miller@pnnl.gov

Dustin Arendt
Pacific Northwest National Laboratory
Dustin.Arendt@pnnl.gov

Abstract

Traditional synonym recommendations often include ill-suited suggestions for writer's specific contexts. We propose a simple approach for contextual synonym recommendation by combining existing human-curated thesauri, e.g. WordNet, with pre-trained language models. We evaluate our technique by curating a set of word-sentence pairs balanced across corpora and parts of speech, then annotating each word-sentence pair with the contextually appropriate set of synonyms. We found that basic language model approaches have higher precision. Approaches leveraging sentence context have higher recall. Overall, the latter contextual approach had the highest F-score.

1 Introduction

Writers often rely on thesauri to recommend synonyms to replace a given word. Though many provide usages of synonyms within example contexts, they cannot account for the specific context of the word "live", i.e., in a writer's specific context. Thus many synonym recommendations are inappropriate, leading writers to reject most synonyms proffered. We propose and evaluate a simple technique to address this problem by combining large scale pre-trained language models and hand-curated thesauri to improve the precision of synonym recommendations in context.

Early research on synonyms and semantic analysis began in the early 1960s and 1970s (Katz and Fodor, 1963) and semantic network representations were popularized by Quillian (Quillian, 1967), Collins (Collins and Quillian, 1969), Woods (Woods, 1975), Brachman (Brachman, 1979) and others. Simplifying these earlier theoretical concepts, the Princeton WordNet was created (Miller et al., 1990). WordNet is a online reference system constructed of synonym sets (synsets) which utilize semantic inheritance to describe word relations. It has served as a widely-adopted baseline tool for synonym retrieval and synset creation.

The NLP community has since developed language models, e.g., Word2Vec (Mikolov et al., 2013), BERT (Devlin et al., 2018) where nearest neighbors in these models' embedding spaces are often synonyms. However, large scale language models have been recently criticized for their complexity, bias, and high energy consumption of training on large-scale web crawls (Bender et al., 2021). Fortunately, pre-trained versions of large scale language models are conveniently available, e.g., SentenceTransformers[1]. This allows researchers or practitioners to use and extend these models for a variety of applications, helping to amortize some costs incurred during training of large scale models.

In addressing the challenges of synonym recommendation and sustainable NLP, this paper makes three contributions: (1) a simple NLP technique using pre-trained language models for contextual synonym recommendation; (2) an approach for producing effective evaluation datasets for this task; and (3) a set of human annotations for the above dataset for our current and future evaluations.

2 Methodology

We identified two baseline synonym tools that rely on token-based lookups — *WordNet* (Miller, 1995) synsets (using the NLTK[2] package (Bird et al., 2009)) and *PyDictionary*[3] (using results from synonym.com) — and an existing counterfactual generative model, PolyJuice (Wu et al., 2021), to provide a baseline comparison for our novel contextual synonym models. Both WordNet and PyDictionary are freely available without API restrictions and commonly used NLP tools, while PolyJuice is a recently released GPT-2 (Radford et al., 2019) model fine-tuned for counterfactual generation.

Two of PolyJuice transformations were relevant: *lexical* and *resemantic* replacement. Lexical re-

[1] https://www.sbert.net
[2] https://www.nltk.org/
[3] pypi.org/project/PyDictionary/

Proceedings of the 2nd Workshop on Simple and Efficient Natural Language Processing, pages 74–78
November 10, 2021.

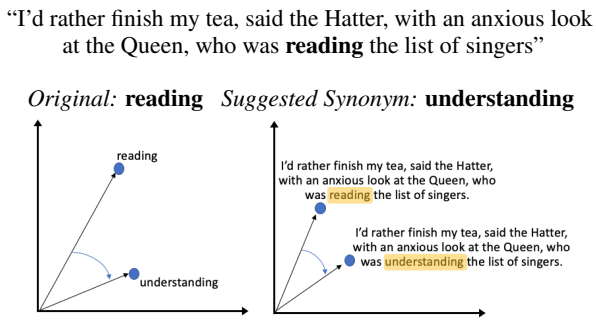

Figure 1: Illustration of the difference calculating cosine similarity between embedding vectors used to filter synonyms for the basic (general) context wrapper (left) and the sentence context wrapper (right) that considers the practical usage in context of the target sentence.

placements suggest alternatives to single words or noun chunks, without altering the part of speech. Resemantic replacements generate replacement phrases that do not alter the dependency tree.

We introduce two novel semantic context wrapper models that leverage pre-trained embeddings to identify synonyms of best fit in context of use. In the *basic context* model, we leverage the general similarity of contextual token embeddings of suggested synonyms to original tokens. This allows us to separate general but unlikely synonyms, edge cases, or noise from the base thesaurus suggestions from those that are generally appropriate in practical use. For example, this is particularly advantageous for tokens that can be used as multiple parts of speech such as "address" or "lead". Our second contextual model, *sentence context*, relies on the distance from the original sentence when the suggested synonym is used in replacement. Both models require a base thesaurus (*e.g.,* WordNet), and we evaluate the performance of three base thesauri: the WordNet and PyDictionary baselines individually, and the union of WordNet and PyDictionary suggestions.

We illustrate the difference in calculating the similarity in the embedding space between the basic (or general) context and sentence context wrappers in Figure 1 for an example sentence: "*I'd rather finish my tea, said the Hatter, with an anxious look at the Queen, who was* ***reading*** *the list of singers*". For both wrappers, we use the same pre-trained sentence-level embedding model, "msmarco-distilroberta-base-v2"[4] using the implementation provided by the sentence_transformers package[5] (Reimers and Gurevych, 2019). The similarity threshold used to exclude unrelated or impractical synonym suggestions is a tunable parameter in our wrapper models, so we establish a high threshold (similarity ≥ 0.3 for basic, 0.9 for sentence context) and low threshold (similarity ≥ 0.25 for basic, 0.8 for sentence context) for each, where thresholds were identified in preliminary experiments that identified tipping points in trade-offs between recall and precision.

3 Evaluation

We curated a balanced annotated dataset of synonyms in context using a vocabulary of candidate synonyms from existing datasets used for synonym research. Our dataset includes examples of their usage in sentences taken from open source datasets covering literary, news, and technical writing.

3.1 Data Preparation

We merged three existing synonym evaluation datasets (SimVerb-3500 (Gerz et al., 2016), SimLex-999 (Hill et al., 2015), and MEN (Bruni et al., 2014)) to build our vocabulary of candidate synonyms. We then identified candidate sentences containing these across our text corpora: *Alice's Adventures in Wonderland* and *The Adventures of Sherlock Holmes*, available from Project Gutenberg[6]; *BBC News*[7], and *NeurIPS* abstracts[8].

We removed candidate words whose synonym sets were smaller than 5 words or greater than 15 then removed words in the bottom 5% and top 95% of the term frequency distribution of the vocabulary. We also excluded sentences with fewer than 80 characters or more than 250 characters, and sentences where spaCy[9] named entity recognition indicated the tagged word for replacement was an entity. This resulted in 735 candidate words across 22,632 candidate sentences.

We ranked candidate words by the total number of different parts of speech (POS) observed across all corpora, and selected the top 35 candidate words. We then randomly selected one sentence from each corpus for each observed POS for a total of 229 sentences to annotate, summarized in Table 1.

[4] https://huggingface.co/sentence-transformers/msmarco-distilroberta-base-v2

[5] https://www.sbert.net/

[6] https://www.gutenberg.org/

[7] http://mlg.ucd.ie/datasets/bbc.html

[8] www.kaggle.com/benhamner/nips-papers

[9] https://spacy.io/

ADJ	ADV	NOUN	VERB	
		x	x	address (6); call (7); care (7); claim (6); drive (6); fall (7); follow (6); hold (8); line (6); map (6); mind (7); note (6); promise (6); reading (7); rise (6); sign (6); stand (6); step (6); stop (7); wait (6); watch (6)
	x	x	x	advance (7)
x			x	narrow (6)
x		x	x	bottom (7); break (7); delay (7); hurt (6); mean (8); post (6); sketch (7); snap (6)
x	x			sure (6); wide (6)
x	x		x	direct (7)
x	x	x	x	lead (9)

Table 1: An overview of the dataset for annotation, where an 'x' indicates the parts of speech observed for the words in their sentence contexts. Parentheses contain the number of sentences the word occurs in.

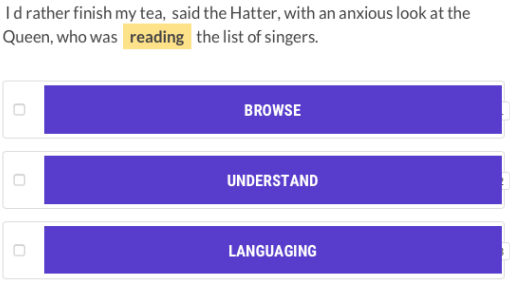

Figure 2: Screen capture of the interface used to collect annotations on synonyms in context. For space, the image has been truncated after the first three synonyms.

3.2 Annotation Protocol

For each sentence, the outputs of our models were merged into a list of synonyms. A Prodigy[10] "text classification recipe" was extended to allow annotators to mark all synonyms that were correct substitutions in the context of the sentence for the given/highlighted word. Figure 2 shows an example of the interface used for annotation.

Authors of this paper were the annotators, so annotation was conducted without indication of which model(s) provided suggestions to avoid bias. We developed and followed the guidelines below:

- Accept if direct substitution of the synonym with the word is appropriate.
- Maintaining syntactic correctness after substitution is not required.
- Given synonyms with different stems, accept only the one with the best substitution.
- Accept synonyms that would be appropriate if they replaced a phrase containing the word.
- Reject words that were part of technical or proper noun phrases.

Overall annotator agreement was high, and annotators took approximately 2 hours to complete the task. We measured agreement by counting the number of annotators who either accepted or rejected a synonym (per word, per context). Annotators "agreed" if the majority of annotators accepted a synonym, or all annotators rejected a synonym. Averaged across all 3911 synonyms, annotators agreed 91% of the time. Averaged across the 782 synonyms where at least one annotator accepted a word, annotators agreed 54% of the time.

3.3 Results

We evaluated model effectiveness using precision, recall, and F-score (F1) calculated per sentence (comparing the set of synonyms produced by the model to those accepted by a majority of annotators), then averaged across sentences. The performance of PolyJuice was very low (F1 < 0.003) relative to the other models, so we omitted it from subsequent analysis and discussion. Figure 3 shows the average precision versus recall. Despite comparable F1, "basic context" had higher precision, outperforming the baselines and other approaches in this dimension, while "sentence context" had higher recall and similarly outperformed the baselines and other approaches. High versus low thresholds affect model performance slightly for basic context, and not at all for sentence context models. Combining WordNet and PyDictionary improved model F1 by greatly improving recall for sentence context models and modestly improved recall for basic context, but did so at the cost of precision.

We also explored whether overall performance was consistent within corpora, show in Fig. 4. These plots reveal that model performance in literary corpora (Alice and Sherlock) was less consistent than the others. Models sHwp and sLwp were within the top three models in each case, only outperformed by bLwp in Sherlock and NeurIPS.

Figure 5 compares the precision and recall of the two model approaches at the individual word level, and shows the same trends seen on aggregate. In cases where the basic context model has non-

[10] https://prodi.gy

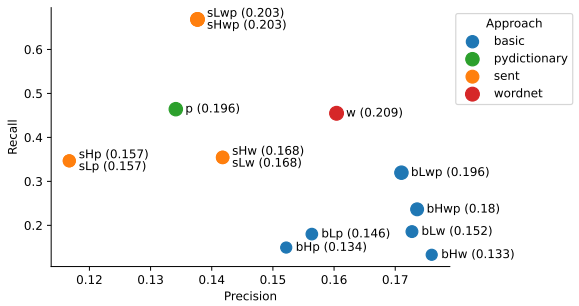

Figure 3: Average Precision versus Recall for top models, colored by model. F-score is shown in parentheses. Model names are abbreviated according to basic versus sentence context, High vs Low threshold, and the combination of WordNet and PyDictionary base.

zero precision/recall, the sentence context model achieves higher recall at the expense of some precision. More often, however, the sentence context model has non-zero precision/recall compared to zero precision/recall for the basic context model.

4 Conclusions and Future Work

Our main finding was that sentence context-based models improve overall performance by trading off a small loss in precision for a large gain in recall. As both techniques were nearest neighbor based, the sentence context approach allowed acceptable synonyms to be identified for the specified context that were less applicable in a general context (i.e., farther away in the word embedding used by the basic context model). An interpretation of this is that pure word embedding techniques will be most effective for the most frequently used synonyms, but sentence context techniques perform better to identify synonyms in less common usage. This is also an area where there is the greatest potential impact for contextual synonym approaches over traditional thesauri or synsets.

We note that overall F-scores were low, indicating the task was challenging and has room for model improvements. For example, a limitation of our approach is the reliance on hand curated synsets or thesauri, i.e., WordNet and PyDictionary. The model's true recall is bounded by the recall of these thesauri. During annotation, there were several cases where none of the recommendations were appropriate, but the annotators remarked they could recall acceptable synonyms. Using proprietary thesauri could be used to improve model performance, e.g., the Miriam-Webster API[11] provides synonyms

[11] https://dictionaryapi.com

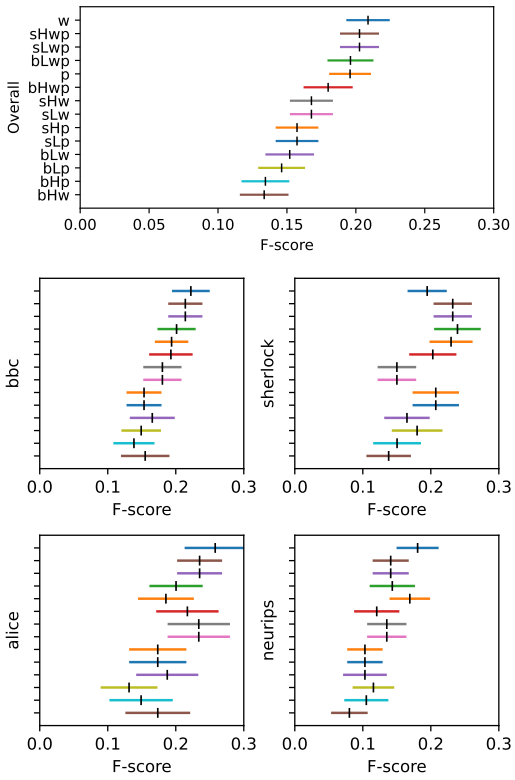

Figure 4: Model performance (F1) overall and within corpora. Error bars show standard error of the mean.

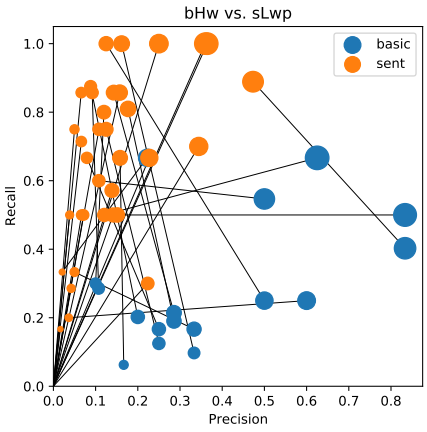

Figure 5: Word-level precision vs. recall for the top models. Lines connect the same word across models.

faceted by verb case as well as contextual sentence examples. Furthermore, limiting context to phrases between the entire sentence and a single word level could further improve performance by eliminating irrelevant sentence context.

Beyond synonym recommendation for end users, our technique could be used to perturb model input to help quantify model robustness, for counterfactual explanations, or for query rewriting.

Acknowledgements

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government or any agency thereof.

References

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Ronald J Brachman. 1979. On the epistemological status of semantic networks. In *Associative networks*, pages 3–50. Elsevier.

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *Journal of artificial intelligence research*, 49:1–47.

Allan M. Collins and M. Ross Quillian. 1969. Retrieval time from semantic memory. *Journal of Verbal Learning and Verbal Behavior*, 8(2):240–247.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Daniela Gerz, Ivan Vulić, Felix Hill, Roi Reichart, and Anna Korhonen. 2016. Simverb-3500: A large-scale evaluation set of verb similarity. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2173–2182.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Jerrold J Katz and Jerry A Fodor. 1963. The structure of a semantic theory. *language*, 39(2):170–210.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

M. Ross Quillian. 1967. Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Science*, 12(5):410–430.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

William A Woods. 1975. What's in a link: Foundations for semantic networks. In *Representation and understanding*, pages 35–82. Elsevier.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.