TopGuNN: Fast NLP Training Data Augmentation using Large Corpora

Rebecca Iglesias-Flores[1] **Megha Mishra**[1] **Ajay Patel**[1] **Akanksha Malhotra**[2]

Reno Kriz[1] **Martha Palmer**[2] **Chris Callison-Burch**[1]

University of Pennsylvania[1] University of Colorado at Boulder[2]

{irebecca, mmishra, ajayp, rekriz, ccb}@seas.upenn.edu

{akanksha.malhotra, martha.palmer}@colorado.edu

Abstract

Acquiring training data for natural language processing systems can be expensive and time-consuming. Given a few training examples crafted by experts, large corpora can be mined for thousands of semantically similar examples that provide useful variability to improve model generalization. We present TopGuNN, a fast contextualized k-NN retrieval system that can efficiently index and search over contextual embeddings generated from large corpora to easily retrieve new diverse training examples. TopGuNN is demonstrated for a semantic role labeling training data augmentation use case over the Gigaword corpus. Using approximate k-NN and an efficient architecture, TopGuNN performs queries over an embedding space of 4.63TB (approximately 1.5B embeddings) in less than a day.

1 Introduction

To collect training data for natural language processing (NLP) models, researchers have to rely on manual labor-intensive methods like crowdsourcing or hiring domain experts. Rather than relying on such techniques, we present TopGuNN, a system to make it quick and easy for researchers to create a larger training set, starting with just a few examples. Large-scale language models can be effectively used to search for similar words or sentences; however, attempting to extract the most similar words from a large corpus can become intractable and time consuming. Our system TopGuNN utilizes a fast contextualized k-NN retrieval pipeline to quickly mine for a diverse set of training examples from large corpora. The system first creates a contextual word-level index from a corpus. Then, given a query word in a training example, it finds new sentences with words used in similar contexts to the query word. Figure 1 shows an example of the results of querying for the word "diagnosis" used in different contexts.

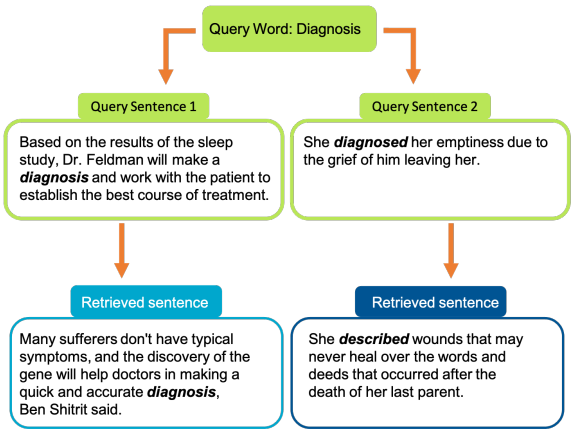

Figure 1: Retrieved results for two queries with different senses of a polysemous word searching over 183 million sentences (or 1.5B embeddings) in the Gigaword corpus with TopGuNN.

TopGuNN pre-computes BERT contextualized word embeddings over the entire corpus, and then efficiently searches through them when queried using approximate k-NN indexing algorithms. Our system has been designed with efficiency and scalability in mind. We demonstrate its use by indexing the Gigaword corpus, a large corpus for which we pre-computed 1.5B contextualized word embeddings (totaling 4.63TB), and using TopGuNN to run search queries over it. A detailed description of the system's architecture is given in Section 3.

1.1 Human-in-the-Loop with TopGuNN

Our primary use case for TopGuNN was to retrieve more training data for an event extraction and semantic role labeling task. We start with a few example sentences of each event type, identify query words within each example sentence (often the event verb), and then query TopGuNN to find new instances of similar sentences. These candidates are quickly voted on by non-expert human annotators who check the correctness of the semantic type (described in Section 2). Using active learning strategies, these filtered candidates can

then be used to better tune TopGuNN's retrieval in the future. We demonstrate how our system can be used to mine for new diverse training data from large corpora with an efficient human-in-the-loop process given just a few samples to start with.

2 Use Case: KAIROS Event Primitives

Our primary use case stems from our work on the DARPA KAIROS program.[1] The DARPA KAIROS program seeks to develop a schema-based AI system that can identify complex events in unstructured text and bring them to the attention of users like intelligence analysts. KAIROS systems are based on an ontology of abstracted event schemas which are complex event templates. Complex event schemas are made up of a series of simpler events, and specify information about participant roles, temporal order, and causal relations between the simpler events. The simplest level event representations used in KAIROS are "event primitives". For each event primitive, a definition of the primitive is given along with the event's semantic roles. An example of a KAIROS event primitive is *Attack*:

Label	**Conflict.Attack**
Description	a violent physical act causing harm or damage
Slot Role	**Slot Argument Constraints**
Attacker	per, org, gpe, sid
Target	loc, gpe, fac, per, com, veh, wea, sid
Instr./Means	com, veh, wea
Place	fac, loc, gpe
Temporal	
Start and End	(times specific to event)
Duration	1 second to multiple years

Each event primitive contained 2-5 example sentences. Prior to TopGuNN, example sentences were selected by linguists who manually retrieved them from a corpus by keyword search. With TopGuNN, we can find thousands of candidate sentences automatically and then annotators can make a quick pass to filter down to the final set.

Some work attempts to create event extraction systems without extensive training data. For instance, Chen et al. (2020) discusses how training could be performed using a single "bleached statement," or a definition of an event, without needing a large set of labeled training examples. Rather than relying on such techniques, we design a system to make it quick and easy for annotators to create a larger training set .

2.1 Corpus

TopGuNN was used to index the Linguistic Data Consortium's English Gigaword Fifth Edition Corpus (Parker et al., 2011). Gigaword consists of approximately 12 gigabytes of news articles from 7 distinct international news agencies, spanning 16 years from 1994-2010, and contains a total of 183 million sentences and 4.3 billion tokens.[2]

2.2 Embedding Model

TopGuNN creates contextualized word embeddings for each content word in the corpus and for each query word in the query sentences. We use BERT (Devlin et al., 2019a) to create the embeddings because BERT produces contextually-aware embeddings unlike word2vec and GloVe (Mikolov et al., 2013; Pennington et al., 2014).[3] FastBERT or DistilBERT would also be appropriate choices, but come with an accuracy trade-off for speed (Liu et al., 2020; Sanh et al., 2019). We also investigated running TopGuNN at the sentence-level using sentence embeddings from SBERT and computing averaged sentence embeddings using BERT (Reimers and Gurevych, 2019). Qualitatively, the results from using BERT at the word-level gave us diversity in the results that we desired (see Appendix B).

2.3 Retrieving Event Primitives

A total of 60 event primitives were annotated using TopGuNN. On average, we were given 2 seed sentences per event and 1-2 viable query words per sentence with which to run through TopGuNN. The query word was typically a verb-form of the event. Approximately 120 query sentences were used to retrieve over 10,000 candidate sentences that were later sent through 2 phases of annotation: 1) sentence classification and 2) span annotation.

After annotators confirm "yes/no" on the candidate sentences meeting the event primitive definition, the sentences classified as "yes" are sent to semantic role labeling for span annotation using a semantic role labeling tool called Datasaur (lee, 2019).[4]

[1] https://www.darpa.mil/program/knowledge-directed-artificial-intelligence-reasoning-over-schemas

[2] https://catalog.ldc.upenn.edu/LDC2011T07

[3] We use the "bert-base-uncased" model from the Transformers Python package. (Wolf et al., 2020)

[4] https://datasaur.ai/

2.4 Examples of Retrieved Sentences

Our system works well in retrieving new, diverse variations of a query word used in contextually similar ways. Below, we display notable retrieved results we found to best showcase the utility of TopGuNN running over the entire Gigaword corpus for gathering both positive and abstract examples for training data.

Positive Example

- Event: Disable
- Definition: Impeding the expected functioning of an ORG, a mechanical device, or software, Ex., remove fuse from explosive

Query Word
The U.S. Army and Marine Corps also both fielded K-9 units with explosive-sniffing dogs to locate IEDs on the battlefield. Engineer Ordinance Disposal (EOD) experts ***disable*** or destroy IEDs through a variety of means, including the use of robotic ground vehicles and explosives.
Retrieved Sentence
Friday's guidelines called for deploying more Patriot interceptor missiles to ***shoot*** down ballistic missiles from North Korea, which has been developing missiles and nuclear weapons.
Cosine sim. of ***disable*** and ***shoot***: 0.641

Table 1: *Shooting down* an explosive as a positive example of *Disable*.

Abstract Example

- Event: Contaminate
- Definition: An animal (incl. people) is infected with a pathogen.

Query Word
"We detected SARS-CoV-2 RNA on eight (36%) of 22 surfaces, as well as on the pillow cover, sheet, and duvet cover," demonstrating that presymptomatic patients can easily ***contaminate*** environments, the authors said. "Our data also reaffirm the potential role of surface contamination in the transmission of SARS-CoV-2 and the importance of strict surface hygiene practices, including regarding linens of SARS-CoV-2 patients," they said.
Retrieved Sentence
Also keep in mind that ***infestations*** of adware/spyware are the leading cause of a slow computer.
Cosine sim. of ***contaminate*** and ***infestations***: 0.637

Table 2: *Infestations* of computer spyware as an abstract example of *Contaminate*.

More notable results can be seen in Appendix C.

2.4.1 Influence of Corpora Size

To validate our system retrieves more relevant results as the size of the corpus it has access to grows we ran a test comparing the results of TopGuNN retrieval on a subset of Gigaword against full Gigaword (see Appendix D). The cosine similarities of retrieved results on the full Gigaword corpus were significantly higher than those retrieved from the subset. Qualitatively, the results appear to contain more apt variations of the retrieved word used in a similar contexts as the query word.

3 System Design

A diagram of TopGuNN is given in Figure 2. TopGuNN is engineered to run in multiple stages: 1) Pre-processing, 2) Generating Embeddings, 3) Indexing, and 4) Running Queries.

3.1 Pre-Processing

During pre-processing we ingest a corpus and perform NLP analysis on each sentence. We use spaCy[5] to generate universal dependency labels and part-of-speech (POS) tags. We use the spaCy annotations to filter down the embeddings to a smaller subset that will be stored and indexed (resulting in a major reduction in the index size).

During pre-processing we also construct several tables in a database to keep track of which sentence and document each word occurs in and what its POS and dependency labels are. This information is stored in 6 lookup dictionaries in a SQLiteDict[6] database seen in Appendix E.

For our use case, we parallelized our preprocessing over each file in Gigaword. In a final step, we amalgamate the 6 lookup dictionaries per file into 6 lookup tables for the whole corpus. By doing so, we were able to use multiple CPUs for pre-processing.

3.2 Generating Embeddings

We partition the 183 million sentences in the Gigaword corpus into 960 sets of approximately 200,000 sentences each. For each partition, we pass batches of 175 sentences through BERT. Each partition is run in parallel using 16 NVIDIA GK210 GPUs on a p2.16xlarge machine with 732GB RAM on AWS, taking approximately 2 days to compute the BERT embeddings for all sentences in Gigaword.

BERT tokenizes its input using the WordPiece tokenization scheme (Devlin et al., 2019b). In TopGuNN, we operate on word-level tokenization

[5] https://spacy.io/

[6] https://pypi.org/project/sqlitedict/

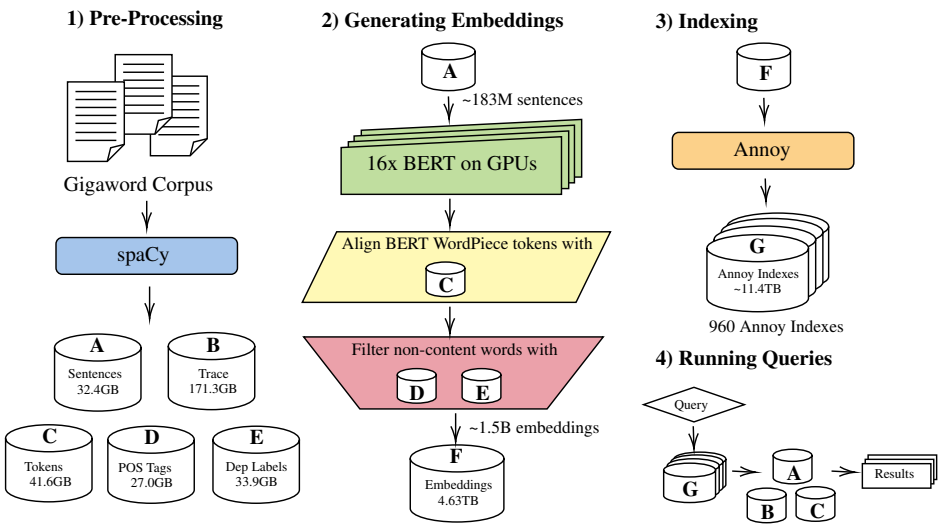

Figure 2: TopGuNN runs in four stages: Pre-Processing, Generating Embeddings, Indexing, and Querying

for indexing and queries, not on word pieces, so we align BERT's WordPiece tokenization scheme to our word-level tokenization scheme. We aligned the BERT-style model's tokenization with spaCy's tokenization using the method described in a blog post by Sterbak (ste, 2018).[7] We then took the mean of the WordPiece embeddings in a word to represent the embedding for the full word.

In order to reduce the number of embeddings we need to store on disk, only content words are kept from each sentence. *Content words* consist of non-proper nouns, verbs, adverbs, and adjectives only. We use POS tags to identify content words and use dependency labels in conjunction with POS tags to further filter out auxiliary verbs. We store the final filtered embeddings using NumPy's memory mapped format as our underlying data store. [8] We discuss the savings in disk space in Section 4.1.

3.3 Indexing

All of the embeddings saved in the previous step for each of the 960 partitions are added to an Annoy index, to create 960 Annoy indexes that span our entire corpus. We use Spotify's Annoy indexing system created by Bernhardsson (2018) for approximate k-NN search, which has been shown to be significantly faster than exact k-NN (Patel et al., 2018). While, there are various competing implementations for approximate k-NN, we ultimately used Annoy to power our similarity search for its ability to build and query on-disk indexes and reduce the amount of RAM required for search.[9]

3.4 Running Queries

TopGuNN allows you to query either a single query word or multiple query words batched together in a search query for performance. The input is a query matrix, which is a matrix of BERT embeddings for all query words in the batch.

Each query word is queried against the 960 Annoy indexes. In order to retrieve the overall top-N results, we query each Annoy index for its top-N results, and we then combine and sort the results from all the Annoy indexes to return the final compiled top-N results. We use our look-up dictionaries to return the document, the sentence, and the word of each result. Search results from each of the query words over the Annoy indexes are combined at the end and exported to a `.tsv` for human annotation and active learning.

3.4.1 Enhancing Query Performance

Sequentially searching each query word against the 960 Annoy indexes before moving on to the next query word is slow. To perform searches more efficiently, we sequentially query each of the 960 Annoy indexes with all query words. This leverages the operating system page cache in such a way that allows for the system to scale better to larger batches of queries. By querying in this manner, we only need to load each of the 960 Annoy index files (each index is ~6GB) into memory once, instead

[7] https://www.depends-on-the-definition.com/named-entity-recognition-with-bert/

[8] https://numpy.org/doc/stable/reference/generated/numpy.memmap.html

[9] https://github.com/spotify/annoy

of once per query word. This is a constant time fixed cost that we must pay for a single query, but subsequent queries will benefit from not having to load the Annoy index again. This fixed cost of loading the Annoy indexes can be amortized over all queries in a batch (see Table 4).[10] Using this method we get performance gains in speed, but we trade it off for higher-memory usage as now we have to hold the intermediate results in memory for all query words in the batch until all Annoy indexes are queried. This means that our memory usage grows linearly with the number of queries in each batch. In practice, we found this trade-off to be tolerable. For a batch of 189 queries, we had a peak memory usage of ~70GB.

3.4.2 Iterative Requery Method

Since this could possibly yield no results if the top-N is sufficiently small and all results are filtered out, we add a parameter that is the number of unique results desired for each query. However, setting top-N to be a very large would hinder the performance of the search queries.

To strike a balance, we employ an iterative requery method that begins with a low top-N and incrementally requeries, increasing N by k (a configurable parameter) while the number of desired unique results retrieved is not met. A current search is halted once the number of desired unique results is met or terminated if the max top-N threshold is reached without meeting the number of desired unique results. This allows us to search the minimum possible amount of nearest neighbors required to reach the best unique results for maximal performance.

4 Performance Details

4.1 Index Size

The size of the Annoy index relies heavily on two parameters set at build time during post-processing: the number of trees (`num_trees`) and the number of nodes to inspect during searching (`search_k`). We also greatly reduce the size of the Annoy index by deciding to exclude non-content words from our index during the Section 3.2 stage.

We use the following heuristic following Patel et al. (2018) to maintain similar search performance across our indexes:

```
num_trees =
max(50,int((num_vecs/3000000.0) * 50.0))
search_k = top_n * num_trees
```

Algorithm 1: Heuristic for Annoy parameters

Excluding Non-content Words We computed the number of words in the entire Gigaword corpus to be 4.3B words. We made the decision to exclude non-content words (defined in Section 3.2) which helped us save resources by a factor of 2.8X while maintaining a high search speed. Using content words only for the Gigaword corpus resulted in a total file size of 16TB (see **F** and **G** in Figure 2).

4.2 Sample Running Times

To give an idea of the TopGuNN system's performance on a corpus as large as Gigaword, we report times for building an index for Gigaword and querying it. Our system design is deconstructed into 4 different stages (as previously described in Section 3) separating out the CPU from the GPU processes in order to streamline the workflow and save on costs. For each stage, we utilized a machine with the best RAM and CPU configuration profile for each particular task and only used a machine with GPUs for Stage 2. For pre-processing, we used a total of 384 cores on a CPU cluster. For our "Generating Embeddings" stage, we utilized a machine with 732GB RAM and 16 GPUs. For post-processing, we used a 16 core machine with 128GB of RAM.

Build Times The times for running the different stages of TopGuNN on the entire Gigaword corpus can be seen in Table 3.

	Build Time
Pre-Processing	76.7 hours
Generating Embeddings	48.8 hours
Post-Processing	23.7 hours

Table 3: Build times for TopGuNN on Gigaword

Query Times The times for querying TopGuNN on the entire Gigaword corpus can be seen in Table 4. The first query word in the batch of queries takes longer as it must load each Annoy index into memory from disk. For subsequent queries in the batch, the Annoy index is already loaded into memory. (see Section 3.4.1)

[10] For example, after searching a query word "identify" on a particular Annoy index all subsequent queried words like "hired" or "launched" on that same Annoy index will leverage the operating system page cache of the Annoy index file and perform faster

	Query Time
Query Batch *(n = 189)*	21.4 hours
First Query *(1)*	19.4 hours
Subsequent Queries *(2-189)*	0.63 minutes

Table 4: Query times for TopGuNN on Gigaword

Because the Annoy indexes are partitioned, the first step could be parallelized to further reduce the 19.4 hours. Keeping cost management in mind, we ran this step serially to highlight its relevant use case even with limited budget (our budget was approximately $2,000).

5 Other NLP Applications

5.1 Sentence- and Document-Level Retrieval

For a sentence-level application, TopGuNN could be useful for training data in story generation. In Ippolito et al. (2020), the author predicts the likely embedding of the next sentence. To facilitate the diversity and speed of candidate sentences used to generate the next sentence in the story, TopGuNN could be employed with sentence embeddings to retrieve sentences from large corpora. For document-retrieval training data, Kriz et al. (2020) recasts text-simplification as a document-retrieval task. The author generates document-level embeddings from the Newsela corpus using BERT and SBERT and similarly adds them to an Annoy index to find documents with similar complexity levels as the query document.

5.2 Multilingual Information Retrieval

DAPRA KAIROS' events are similar to the events found in the IARPA BETTER multilingual information retrieval project.[11] A future application of TopGuNN could be querying in English and retrieving training examples in another language (or vice versa) by substituting BERT for GigaBERT (Lan et al., 2020) in TopGuNN. With this modification, TopGuNN could help facilitate multilingual retrieval of training examples.

6 Related Work

Previous work that parallels our work to search and index large corpora includes projects like Lin et al. (2010), which created an index of n-gram counts over a web-scale sized corpus. Similarly, as an extension to work completed by Lin et al. (1997) and Gao et al. (2002), Moore and Lewis (2010) propose a method for gathering domain-specific training data for languages models for use in tasks such as Machine Translation. By utilizing contextual word embeddings from a modern language model like BERT instead of techniques like n-grams or perplexity analysis as seen in previous approaches, TopGuNN aims to achieve higher quality results.

Our work directly builds upon prior research on approximate k-NN algorithms for cosine similarity search. We chose to use the Annoy package for indexing our embeddings in TopGuNN for its particular ability to build on-disk indexes, however, another package could be used instead. Aumüller et al. (2018) discusses various approximate k-NN algorithms that could alternatively be utilized for TopGuNN with alternate trade-offs in speed, memory, and other hardware requirements. By utilizing on-disk indexes on SSDs, which have fast random-access reads and high-throughput, we are able to use significantly cheaper machines than would be required to hold terabytes of indexes in RAM.

7 Getting started with TopGuNN

You can get started with TopGuNN on GitHub: https://github.com/Penn-TopGuNN/TopGuNN

8 Conclusion

We have presented a system for fast training data augmentation from a large corpus. To the best of our knowledge, existing search approaches do not make use of contextual word embeddings to produce the high quality diverse results needed in training examples for tasks like our event extraction use case. We have open sourced our efficient, scalable system that makes the most efficient use of human-in-the-loop annotation. We also highlight several other NLP tasks where our system could facilitate training data augmentation in Section 5.

Future work may include enabling TopGuNN to query for multi-word expressions (i.e. *"put a name to"*), hyphenated expressions (i.e. *"pre-existing conditions"*), or in the form of natural language questions as seen in (Yu et al., 2019). Finally, identifying antonymy as studied in (Rajana et al., 2017) would be a valuable extension for more fine-grained search results as synonyms and antonyms often occupy the same embedding space.

[11] https://www.iarpa.gov/index.php/research-programs/better

Acknowledgements

We would like to thank Erik Bernhardsson for the useful feedback on integrating Annoy indexing.

Special thanks to Ashley Nobi for spearheading the annotation effort and Katie Conger at University of Colorado at Boulder for the training sessions on semantic role labeling she gave for the span annotation effort.

We would like to thank the Fall 2020 semester students of *CIS 421/521 - Artificial Intelligence* and Leila Pearlman at the University of Pennsylvania, and the University of Colorado at Boulder's Team of Linguists for annotating TopGuNN results.

We would like to thank Ivan Lee, CEO of Datasaur Inc., Hartono Sulaiman and Nadya Nurhafidzah of Datasaur, for providing a seamless annotation tool for us to use and with around-the-clock customer service in navigating the system.

I would like to thank my post-doc Dr. Mohammad Sadegh Rasooli and my PhD labmate Aditya Kashyap for their invaluable input and constant availability to us throughout the project.

Special thanks to my senior PhD labmate Reno Kriz for his mentorship during this project.

The first author was funded by NSF for the University of Pennsylvania under grant number DGE-1845298 (the Graduate Research Fellowships Program). This research is also supported in part by the DARPA KAIROS Program (contract FA8750-19-2-1004), the DARPA LwLL Program (contract FA8750-19-2-0201), and the IARPA BETTER Program (contract 2019-19051600004). Approved for Public Release, Distribution Unlimited. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, IARPA, or the U.S. Government.

The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of NSF, DARPA, and the U.S. Government.

And to the timeless 1986 American cult-classic "Top Gun," thanks for the inspiration on naming our retrieval system... ***I feel the need for speed!***

References

2018. *Named entity recognition with Bert*. Tobias Sterbak Consulting, Akazienstraße 3A, 10823 Berlin, Germany.

2019. *Ivan Lee, CEO., Datasaur*. Datasaur, Inc., Sunnyvale, California, United States.

Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. 2018. Ann-benchmarks: A benchmarking tool for approximate nearest neighbor algorithms.

Erik Bernhardsson. 2018. *Annoy: Approximate Nearest Neighbors in C++/Python*. Python package version 1.13.0.

Yunmo Chen, Tongfei Chen, Seth Ebner, Aaron Steven White, and Benjamin Van Durme. 2020. Reading the manual: Event extraction as definition comprehension.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jianfeng Gao, Joshua Goodman, Mingjing Li, and Kai-Fu Lee. 2002. Toward a unified approach to statistical language modeling for chinese. *ACM Transactions on Asian Language Information Processing*, 1(1):3–33.

Daphne Ippolito, David Grangier, Douglas Eck, and Chris Callison-Burch. 2020. Toward better storylines with sentence-level language models.

Reno Kriz, Eleni Miltsakaki, Jaime Rojas, Rebecca Iglesias-Flores, Megha Mishra, Marianna Apidianaki, and Chris Callison-Burch. 2020. Recasting text simplification as a document retrieval task. *In Submission*.

Keita Kurita, Nidhi Vyas, Ayush Pareek, Alan W Black, and Yulia Tsvetkov. 2019. Measuring bias in contextualized word representations.

Wuwei Lan, Yang Chen, Wei Xu, and Alan Ritter. 2020. An empirical study of pre-trained transformers for arabic information extraction.

Dekang Lin, Kenneth Ward Church, Heng Ji, Satoshi Sekine, David Yarowsky, Shane Bergsma, Kailash Patil, Emily Pitler, Rachel Lathbury, Vikram Rao, et al. 2010. New tools for web-scale n-grams. In *LREC*.

Sung-Chien Lin, Chi-Lung Tsai, Lee-Feng Chien, Keh-Jiann Chen, and Lin-Shan Lee. 1997. Chinese language model adaptation based on document classification and multiple domain-specific language models. In *EUROSPEECH*. ISCA.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Haotang Deng, and Qi Ju. 2020. Fastbert: a self-distilling bert with adaptive inference time.

Tomas Mikolov, Kai Chen, G. S. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR*.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden. Association for Computational Linguistics.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. *English Gigaword Fifth Edition LDC2011T07*. Web Download.

Ajay Patel, Alexander Sands, Chris Callison-Burch, and Marianna Apidianaki. 2018. Magnitude: A fast, efficient universal vector embedding utility package. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 120–126, Brussels, Belgium. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Yusu Qian, Urwa Muaz, Ben Zhang, and Jae Won Hyun. 2019. Reducing gender bias in word-level language models with a gender-equalizing loss function. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 223–228, Florence, Italy. Association for Computational Linguistics.

Sneha Rajana, Chris Callison-Burch, Marianna Apidianaki, and Vered Shwartz. 2017. Learning antonyms with paraphrases and a morphology-aware neural network. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 12–21, Vancouver, Canada. Association for Computational Linguistics.

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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2019. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task.

A Ethical Considerations/Discussion

Our work utilizes BERT and therefore it contains the inherent biases that exist in language models trained on large amounts of unsupervised data collected from the internet. Kurita et al. (2019) analyzes the various biases that exist specifically in BERT.

In our own tests, we directly observed some of these biases when querying for the DARPA KAIROS DETONATE:EXPLODE event over a subset of Gigaword. Querying the word **bombing** in the sentence *"Rabee'a owned a drill rig, and his friend had heard stories from elsewhere in Yemen about jets **bombing** well sites."* yielded the word **Muslim** as the top result from the sentence *"Amid the tension, **Muslim** leaders say their communities are doing more than ever to help in investigations – a point they say is overlooked by many Americans."* with a cosine similarity of 0.602. Moreover, 9 out of the 20 top results were the words "muslim" or "mosque".

When using TopGuNN to help bootstrap training data for event extraction models or running search queries, care must be taken to ensure these biases do not leak into a downstream applications by a thorough manual review to prevent unintentional harm. Debiasing language models is an active area of research and techniques like Qian et al. (2019) could be utilized to attempt to debias language models at train time that could then replace BERT in TopGuNN.

B Testing Various Embedding Models with TopGuNN

We explored 3 different embedding models for the TopGuNN system:

1. SBERT automatically has its own sentence representation to retrieve sentences.
2. AVG-BERT uses the mean of the word embeddings as the sentence representation to retrieve sentences.
3. BERT returns results for a single query word and retrieves sentences with words that were used in a similar context as the query word in the query sentence.
   Note: To show diversity of results for BERT, the Top-10 unique nearest neighbors are shown and not necessarily the first Top-10 as seen in SBERT and AVG-BERT.

Table 5: Results comparing SBERT, AVG-BERT, and BERT

Method	Cosine Sim	Retrieved Result
Query Sentence: "President Barack Obama's hopes of winning Senate approval for a new arms control treaty with Russia by the end of the year were encouraged Tuesday by two Republican senators, including John McCain."		
	0.908	WASHINGTON -The US Senate, in a key test vote, moved Tuesday toward final passage of a nuclear arms pact with Russia, setting up a likely foreign policy victory for President Obama and a hard-won achievement for Senator John F. Kerry of Massachusetts, who shepherded the treaty through fierce GOP opposition.
	0.888	Fresh from winning Senate approval for a new strategic arms treaty, President Barack Obama plans to return to the negotiating table with Russia next year in hopes of securing the first legal limits imposed on the smaller, battlefield nuclear weapons viewed as most vulnerable to theft or diversion.

Continued on next page

Table 5 – continued from previous page

Method	Cosine Sim	Retrieved Result
SBERT	0.882	McCain, one of his party's leading voices on national security, said he thought that Republican concerns over missile defense and nuclear modernization could be resolved in time to vote on the so-called New Start treaty during the lame-duck session of Congress this month, as Obama has sought.
	0.878	To press their point, Republicans pushed through a side resolution calling on Obama to open talks with Russia on such weapons within a year.
	0.872	The Senate moved closer Monday to approving a new arms control treaty with Russia over the opposition of Republican leaders as lawmakers worked on a side deal to assure skeptics that the arms pact would not inhibit U.S. plans to build missile defense systems.
	0.872	Beyond his behind-the-scenes role in negotiating the tax deal with Republicans – a path that Biden and Obama decided on in a recent conversation at the White House, aides say – the vice president has also been trying to win Republican votes in the Senate for ratification of the START nuclear arms treaty with Russia.
	0.856	On Tuesday, Sen. John McCain – who is inexplicably playing second fiddle to Kyl – told ABC: "I believe that we could move forward with the START treaty and satisfy Senator Kyl's concerns and mine about missile defense and others, and I hope that we can do that."
	0.848	White House officials, meanwhile, expressed hope of sealing a deal swiftly, perhaps by midweek, and clearing the congressional calendar for a long list of other priorities they aim to accomplish by the end of the year, including ratification of the New START arms treaty with Russia and the repeal of the "don't ask, don't tell" policy for gay service members as part of a wider Pentagon policy bill.
	0.832	While President Barack Obama presses the Senate to embrace a new arms control treaty with Russia, another nuclear pact with Moscow secured final approval after more than four years on Thursday with virtually no notice but potentially significant impact.
	0.828	In the interview, Putin also warned that Russia would develop and deploy new nuclear weapons if the United States did not accept its proposals on integrating Russian and European missile defense forces – amplifying a comment made by Medvedev in his annual state of the nation address Tuesday.
	0.920	WASHINGTON - Senator John F. Kerry and other top Democrats said Tuesday they have secured enough bipartisan backing to ratify the START nuclear arms treaty with Russia, a vote that would be a substantial foreign policy victory for President Obama.
	0.911	Immediately after the tax vote Wednesday, Senate Democrats began angling for passage of a new U.S.-Russian nuclear arms treaty, a priority of President Barack Obama that has been on the agenda for months.
	0.904	McCain, one of his party's leading voices on national security, said he thought that Republican concerns over missile defense and nuclear modernization could be resolved in time to vote on the so-called New Start treaty during the lame-duck session of Congress this month, as Obama has sought.

Continued on next page

Table 5 – continued from previous page

Method	Cosine Sim	Retrieved Result
AVG-BERT	0.896	Sen. John McCain of Arizona had previously said he hoped to vote for the treaty as long as concerns over missile defense were addressed, and it was not clear whether he was signaling a shift or using the opportunity to vent his longstanding frustration with Russian behavior.
	0.894	A Republican senator announced that he would vote for the treaty and two others said they were leaning toward it, and at the same time, Sen. John McCain, R-Ariz., produced separate legislation that could reassure fellow Republicans worried about the treaty's impact on missile defense.
	0.893	Obama brought up the treaty Tuesday during a White House meeting with congressional leaders, pressing them to vote this month to strengthen the relationship with Russia.
	0.892	President Barack Obama on Tuesday strongly defended his tax cut deal with congressional Republicans against intense criticism from his own party, insisting it was "a good deal for the American people."
	0.887	Sen. Harry Reid of Nevada, the majority leader and crucial proponent of the repeal, noted that some Republicans had indicated they may try to block Senate approval of a nuclear arms treaty with Russia due to their pique over the Senate action on the ban on gays in the military.
	0.887	Obama has insisted that the Senate approve it before the end of the month rather than wait until a new Senate with more Republicans takes office, and a number of Republican senators have signaled tentative support.
	0.885	Obama, in his brief remarks Wednesday during a meeting with the president of Poland, suggested that Republicans for the next two years will still be defending the Bush tax rates while he is looking forward to a new, better code.
Query Word: "President Barack Obama's hopes of ***winning*** Senate approval for a new arms control treaty with Russia by the end of the year were encouraged Tuesday by two Republican senators, including John McCain."		
BERT	0.953	TWO REPUBLICANS HINT AT HOPE FOR ARMS PACT WITH RUSSIA President Barack Obama's hopes of ***winning*** Senate approval for a new arms control treaty with Russia by the end of the year were encouraged Tuesday by two Republican senators, including John McCain.
	0.825	Obama's failure to ***win*** passage of comprehensive immigration reform was a disappointment to many Latinos, he conceded.
	0.802	Aides to Reid said they had mapped out a path to ***securing*** votes on all of the legislation, which would mean staying in session until next Thursday, two days before Christmas, and potentially returning the week before New Year's Day.
	0.793	While he has a fair chance of ***securing*** the votes of the two other Democrats, he faces a potential fight with one of those commissioners, Michael J. Copps, who has been public in his support for stricter regulation of broadband Internet service.
	0.756	With a week before Election Day, Perry, who is thought to have the best chance of ***gaining*** a seat for Republicans in the state, is struggling to fend off accusations that he witnessed and covered up the illegal strip search of a teenage girl in 1991, when he was a police sergeant in Wareham, Mass.

Continued on next page

Table 5 – continued from previous page

Method	Cosine Sim	Retrieved Result
	0.755	The White House is negotiating with Sen. Jon Kyl, R-Ariz., whose support is crucial to ***getting*** other Republican votes, to meet his price: more money to modernize the nuclear arsenal.
	0.744	Republican confidence about ***capturing*** control of the House remained high, though even Republicans considered the Senate more of a question mark, given the number of excruciatingly close races across the country.
	0.731	Obama bested the chamber in the first two years of his term, ***passing*** health care legislation and an overhaul of financial regulations over the group's heated opposition.
	0.731	Like most of her 18 opponents nearing the Nov. 28 election, Manigat's campaign trail stretches northward from Port-au-Prince to Miami, New York, Boston and Montreal in hopes of ***garnering*** money and influence from the large Haitian diaspora.
	0.730	Still, with Republicans ***challenging*** every element of the new law, the Obama administration is likely to be handcuffed in its efforts to expand the revamping of the health care system.

C Notable Results

Positive Example

- Event: Defeat
- Definition: Defeat in a conflict or an election (but not a game-style competition)

Query Word
Most democratic activists and lawmakers ***rejected*** the deal as a sham and it was eventually defeated in the city's legislatures after a botched walkout by pro-government legislatures.
Retrieved Sentence
The White House and Senate Democrats ***considered*** the amendment a treaty killer because any change to the text would require both countries to go back to the negotiating table.
Cosine sim. of ***rejected*** and ***considered***: 0.745

Table 6: *Treaty killer* as a positive example of *Defeat*.

Positive Example

- Event: Disable
- Definition: Impeding the expected functioning of an ORG, a mechanical device, or software, Ex., remove fuse from explosive

Query Word
Soldiers and personnel have to be trained to be aware of the enemy's behaviors, to look for indicators of IEDs in their patrol areas and to use technology to dispose or ***disable*** them.
Retrieved Sentence
And he assured his audience that he had made clear to senior Pakistani military officials my strong desire to see more action taken against these places and to ***root*** out the terrorists.
Cosine sim. of ***disable*** and ***root***: 0.616

Table 7: *Rooting out terrorist* organizations as a positive example of *Disable*.

Positive Example

- Event: Block Passage
- Definition: Preventing entry or exit from a location

Query Word
...archipelagic defense would have the holders of islands adjoining straits and other narrow seas fortify those islands with mobile anti-ship and anti-air missiles while deploying surface, subsurface, and aerial assets to ***block*** passage through these seaways. In effect these forces string a barricade between geographic features—interdicting shipping and overflight while bringing economic and military pressure on adversaries.
Retrieved Sentence
Assad Ismail, a local council president in Sadiya, a village along the disputed territories northeast of Baghdad, said that only the Americans were able to settle a recent dispute that flared when Iraqi soldiers trying to ***restrict*** the movement of insurgents closed off local farmers' access to their date palms, tomatoes and peanuts.
Cosine sim. of ***block*** and ***restrict***: 0.637

Table 8: *Restricting movement* as a positive example of *Block Passage*.

Positive Example

- Event: Destroy
- Definition: Damage property, organization or natural resource

Query Word
"These actions challenge national sovereignty, threaten one country, two systems, and will ***destroy*** the city's prosperity and stability," she said, referring to slogans of "Liberate Hong Kong, revolution of our times" and the act of throwing a Chinese flag in the sea.
Retrieved Sentence
"Letting it expire would threaten jobs, harm the environment, ***weaken*** our renewable fuel industries, and increase our dependence on foreign oil," they wrote.
Cosine sim. of ***destroy*** and ***weaken***: 0.732

Table 9: *Weaken renewable fuel* as a positive example of *Destroy*.

Abstract Example

- Event: Destroy
- Definition: Damage property, organization or natural resource

Query Word
"These actions challenge national sovereignty, threaten one country, two systems, and will ***destroy*** the city's prosperity and stability," she said, referring to slogans of "Liberate Hong Kong, revolution of our times" and the act of throwing a Chinese flag in the sea.
Retrieved Sentence
Adopting an orthodox view, he said in 1976 that a projected budget deficit estimated at 60 billion was "very scary" and would "***wreck***" the economy.
Cosine sim. of ***destroy*** and ***wreck***: 0.752

Table 10: *Wrecked the economy* as an abstract example of *Destroy*.

Abstract Example

- Event: Block Passage
- Definition: (Physically) preventing entry or exit from a location

Query Word
...archipelagic defense would have the holders of islands adjoining straits and other narrow seas fortify those islands with mobile anti-ship and anti-air missiles while deploying surface, subsurface, and aerial assets to ***block*** passage through these seaways. In effect these forces string a barricade between geographic features—interdicting shipping and overflight while bringing economic and military pressure on adversaries.
Retrieved Sentence
Even as Pakistan's army vows to take on militants spreading chaos and mayhem inside Pakistan, the intelligence service still sees the Afghan Taliban as a way to ensure influence on the other side of the border and ***keep*** India's influence at bay.
Cosine sim. of ***block*** and ***keep***: 0.649

Table 11: *Keeping influence at bay* as an abstract example of *Block Passage*.

D TopGuNN Results Using Different Sized Corpora

We compared the top 10 unique results from a small subset of the Gigaword corpus (400,000 sentences) compared to results ran on the full Gigaword corpus (183 million sentences) for the event primitive **Sentence** (as in the judicial meaning).

Current findings have shown us some interesting, but unexpected results. The cosine similarities of retrieved results for full Gigaword are significantly higher, but TopGuNN still works extremely well on a small subset in terms of quality and diversity of results. Other researchers who need to prioritize high-speed in retrieving positive or abstract examples for their training data could retrieve similar sentences even faster on a smaller subset of a uniform corpus like Gigaword without having to sacrifice much in terms of quality.

Table 12: Top-10 unique results querying the event primitive 'Sentence' (as in the judicial meaning) over a subset of Gigaword (400K sentences) vs. full Gigaword (183M sentences).

Method	**Cosine Sim**	**Retrieved Result**
Query Sentence: "The judge ***sentenced*** him to death."		
Gigaword Subset *(400K sentences)*	0.742	When she explained to the court that she could not afford to pay, Nowlin was ***sent*** to prison.
	0.701	"It matters little if they ***condemn*** me, even to the heaviest sentence.
	0.695	True, the court could have gone further and actually ***jailed*** the two defendants.
	0.693	He ***received*** a life sentence.
	0.680	A federal judge ***spared*** him prison time but ordered him to leave the country within 90 days or be deported.
	0.676	The jury came within two votes of ***convicting*** Megahed.
	0.670	Bush ***commuted*** the sentences, and the men are now free.
	0.666	At 6:28 p.m., she found him ***hanged***.
	0.664	He eventually pleaded guilty to manslaughter and ***spent*** 15 years in prison.

Continued on next page

Table 12 – continued from previous page

Method	Cosine Sim	Retrieved Result
	0.659	That convinced a jury to find him guilty of aggravated sexual assault and ***send*** him to prison for 75 years.
Full Gigaword *(183M sentences)*	0.881	A jury didn't believe him, and a ***judge*** sentenced him to eight years in prison.
	0.863	Seven years later, a Paris court ***condemned*** him in absentia to life in jail for the murders.
	0.847	The jury decided unanimously to ***sentence*** him to death.
	0.845	The jury ***convicted*** him to life in prison, where he will spend the rest of his life.
	0.841	"There are some things you just can't run from, this being one of those," Rolling told Circuit Judge Stan R. Morris, who accepted the pleas and found him guilty and ***later*** sentenced him to death.
	0.833	The presiding judge agreed, ***sentencing*** the two young men to life imprisonment.
	0.830	The judge ***sent*** them to prison.
	0.828	It wasn't until last October – a decade later – that ***courts*** sentenced 34 men to 26 years each for the killings.
	0.814	They unanimously ***acquitted*** him on all counts.
	0.812	But the U.N. court decided he was not directly involved and ***punished*** him with a light two years in prison.

E Lookup Dictionaries

1. Sentences (32.4GB):
   `sent_id → (sentence)`
2. Document Traceability (15.0GB):
   `sent_id → (doc_id)`
3. Tokens (41.6GB):
   `sent_id → (sentence tokens)`
4. Parts-of-Speech Tags (27.0GB):
   `sent_id → (sentence pos_tags)`
5. Dependency Labels (33.9GB):
   `sent_id → (sentence dep_labels)`
6. Words Trace (156.3 GB):
   `word_id → (word_id, word, (doc_id, sent_id))`

F Querying Polysemous Words

We demonstrate TopGuNN's ability to perform contextual similarity search of a query word in its corresponding sentence using polysemous words, which have two distinct sentences. Figure 3 and Figure 4 are further examples of querying two distinct sentences with different senses of the same word to retrieve sentences that capture both polysemies.

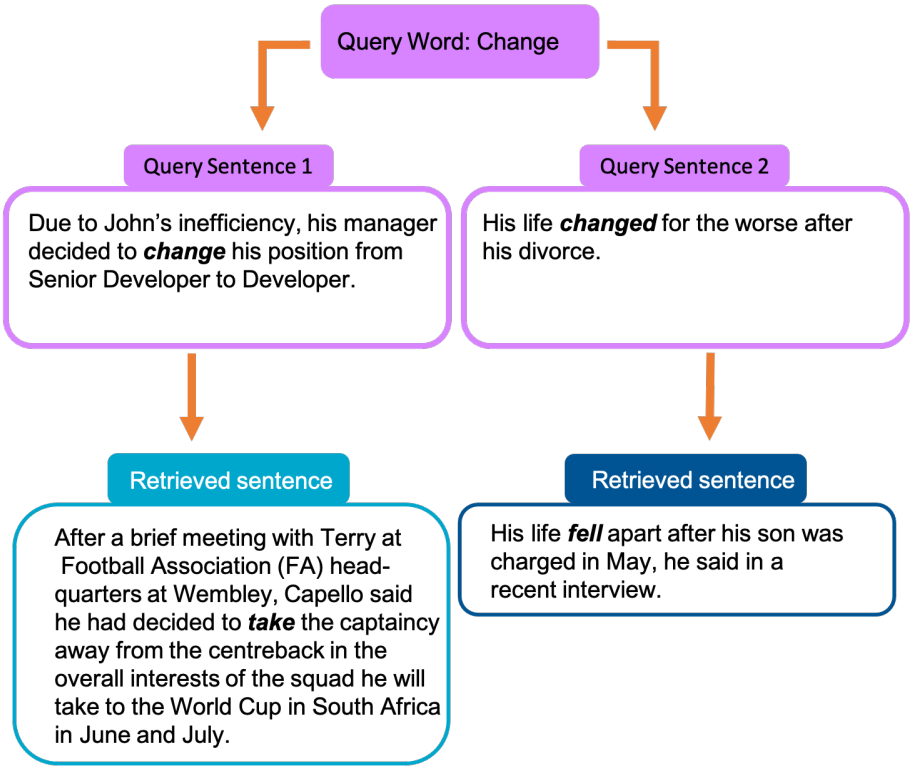

Figure 3: TopGuNN results on the the polysemous word *change*.

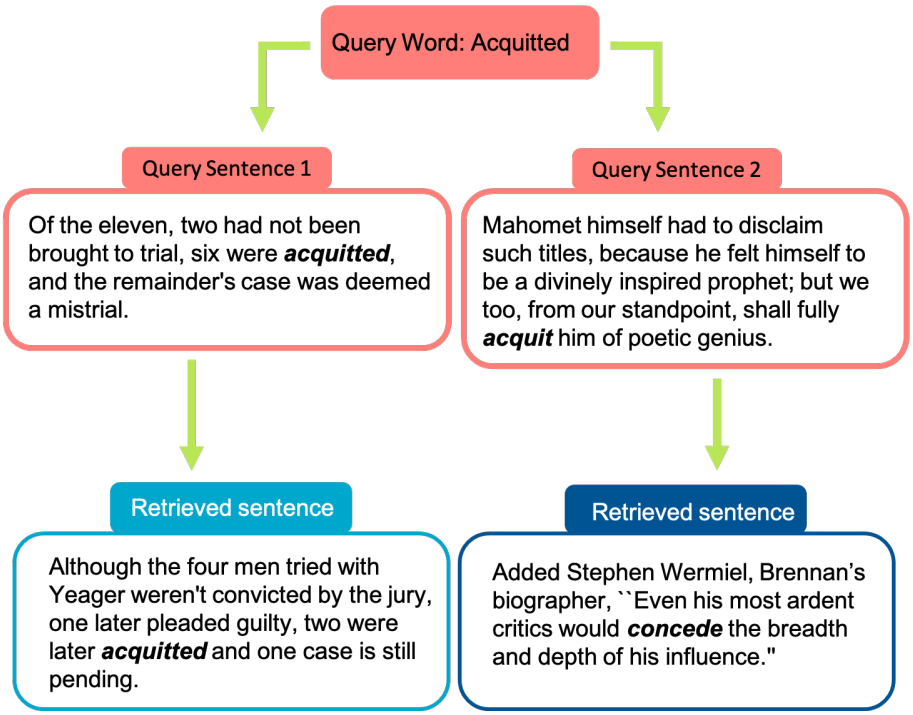

Figure 4: TopGuNN results on the the polysemous word *acquit*.