Improved Semantic Search based on Weighted TF-IDF & BERT

Yuxiang Li
University of Illinois at Urbana-Champaign
Champaign, Illinois
yl48@illinois.edu

Yuxin Xiao
University of Illinois at Urbana-Champaign
Champaign, Illinois
yuxinx2@illinois.edu

Zhen Fan
University of Illinois at Urbana-Champaign
Champaign, Illinois
zhenfan3@illinois.edu

ABSTRACT

We present an improved semantic search approach based on a weighted TF-IDF method and the BERT natural language model. We motivate the choice of a weighted TF-IDF method via an intuition that the questionable spans in a document summarize the document's topics and hence, should be placed greater emphasis when calculating the TF-IDF score. The use of the BERT natural language model is to complement the weakness of the TF-IDF framework in understanding the true semantic meaning of a document. Therefore, our model encodes a document's questional spans and true semantics. It scales effectively in the size of the dataset. In a number of semantic search experiments on question-answering datasets, we demonstrate that our approach outperforms existing related methods by a significant margin.

1 INTRODUCTION

1.1 Questionable Spans

Reading and writing make up a crucial component in daily communication in the human world. People exchange information and share their options through this process. On the one hand, when an author writes a piece of text, he would like to emphasize on a certain topic and offer relevant information to this topic. On the other hand, when a reader reads a piece of text, he wants to find out the author's focus and get more details about this focus.

Typically, a document's topic and its relevant details provide all the useful information in the document. However, in order to make the document follow a human-readable format, people need to use many meaningless words to join meaningful pieces of text together. As a result, noise is introduced into the document via those meaningless words in this process. The noise confuses many existing information retrieval models and gives rise to topic misclassification.

We would like to define a novel concept, "Questionable Spans", which corresponds to the parts in a sentence that stand a high chance of forming the answer to a potential question targeting the original sentence. For example, consider a simple sentence, "the final exam is on Wednesday". In this sentence, "exam" is the subject and "Wednesday" tells when the action happens. Hence, questionable spans in the given sentence refer to these two words as they are more likely to become answers to target questions and the rest are devoid of useful information.

Therefore, the questionable spans in a document give the whole details about the topic of the document and formulate the answer pool to all the potential questions targeting the document. If we can identify all these spans, then we are able to capture all the meaningful information in the document and better represent the document with these spans.

1.2 User Query Model

In the case of semantic search, when a user inputs a query, we would like to find the most semantically relevant document from our database and extract useful information as the answer to the query. In order to go through this find and extraction process effectively, it will be beneficial if we can preprocess the documents in the database.

Consider the questionable spans discussed in the previous subsection. These spans form an answer pool to all the possible questions targeting the documents in the database. Consequently, it is highly possible that the useful information we would like to extract as an answer comes from these spans. Therefore, we wish to identify all the questionable spans in all the documents in the database in advance and give them a higher weight when calculating the TF–IDF score.

The higher weight given to the questionable spans allow those spans to represent the documents to a greater extent and diminish the noise associated with the non-questionable parts. In fact, by preprocessing all the questionable spans, we try to exhaustively find out all the questions that could be asked based on our documents in the database. In this way, we can quickly match the user query to the questions we gathered in advance and simply provide the corresponding questionable spans as the answer to the user query.

1.3 BERT Natural Language Model

In contrast to the TF–IDF framework which makes use of the word's frequency as the measure of its discriminative ability, the BERT (Bidirectional Encoder Representations from Transformers) natural language representation model [**?**] makes use of Transformer, an attention mechanism which reads the entire sequence of words at once. Therefore, the model is considered bidirectional and this characteristic allows the model to the learn the contextual relations between words (or sub-words) in a text based on all of its surroundings (left and right of the word).

It has been proved that the BERT model is more capable of understanding the true semantic meaning of the text by presenting state-of-the-art results in a wide variety of Natural Language Processing tasks, including Question Answering (SQuAD v1.1), Natural Language Inference (MNLI), and others.

Hence, we would like to incorporate the BERT model in our approach so as to complement the limitation of the TF–IDF framework in comparing the true semantic meaning of the user input query and the questionable spans in selected documents.

1.4 Paper Organization

The rest of the paper is organized as follows. We begin by introducing the problem we would like to address in Section 2. We then

discuss how our work is connected with the existing work in the field of semantic search in Section 3. In Section 4, we explain our approach with sufficient details. In Section 5, we outline the experiment setup and evaluate the experiment results. In Section 6, we summarize our work, draw conclusions and discuss how our work can be further improved in the future.

2 PROBLEM STATEMENT

Semantic search seeks to improve the search precision by understanding a searcher's intent through contextual meaning. Through concept matching, synonyms, and natural language models, semantic search provides more interactive search results through transforming structured and unstructured data into an intuitive and responsive database. Semantic search brings about an enhanced understanding of searcher intent, the ability to extract answers, and delivers more personalized results.

We would like to address the problem of semantic search by improving the precision of the search results. More specifically, when a user inputs a query, we want to find the top 10 most relevant sentences that can be used as the answer to the user query. Among these 10 provided sentences, we want to rank them according to their relevance to the user query and reduce the number of semantically irrelevant sentences.

However, the traditional TF–IDF method only uses a word's frequency to infer its discriminative score and simply ignores its real semantics and significance level in the text. We would like to improve the performance of the TF–IDF framework in semantic search by allowing the TF–IDF method to pay greater attention to the questionable spans in the document and to work with the BERT natural language model to reduce semantic ambiguity.

3 RELATED WORK

Some existing related works in the area of semantic search extend the TF–IDF framework in various ways. However, they are limited in terms of the model efficiency and not addressing the significance of the questionable spans in the document. **?** [**?**] applied the K-nearest neighbor model to aid the categorization of the document according to its distance to the training documents. They then incorporated this categorization information into the calculation of the TF–IDF score. However, by doing so, they assumed that each document only contained one single topic and simply ignored the potential subtopics which could become answers to user queries. **?** [**?**] expanded the queries by considering "keyword + tags" instead of keywords only when measuring the TF (term frequency). This approach required manually adding tags to documents, which involved a lot of work and was less efficient. **?** [**?**] proposed to use Improved Gini Index algorithm instead of the IDF part in the TF–IDF algorithm. **?** [**?**] used Shannon's word entropies provided by the TF–IDF transform to reweigh the word embeddings. Nonetheless, these two propositions isolated the words and did not take the contextual relations between words in a text into consideration.

Other vector space modeling approaches [**? ? ?**] in the field of semantic search suffer from problems like the poor representation of long documents, the lack of semantic sensitivity, and the loss of order in which terms appear in the document.

Natural language models like the BERT model [**?**] require a rather large amount of computation work during training and are slow when calculating the semantic similarities. When working alone, these models are not able to provide answers to user queries in time or handle the case of inputting an incomplete sentence.

To summarize, our proposed approach excels at providing relevant and useful answers to user queries effectively and efficiently. It considers different scenarios of users' inputs and the contextual relations between words. It highlights the questionable spans in the documents and quickly generates ranked answers to user queries.

4 APPROACH

4.1 General Framework

Our general framework is described in Figure 1. With a new query, we first use a syntax analysis tree (see section 4.2) to detect whether it is a sequence of words or a complete sentence. This is for later applying BERT model in hope of semantically better results if the query is a complete sentence. If a query is a sequence of words, we use Weighted TF–IDF scores to rank our final results; if it is a complete sentence, we use Weighted TF–IDF first for a larger number of potential results, then rank the results by the similarity of the query vector and the vectors for each potential result, which are generated from the BERT model (see section 4.6).

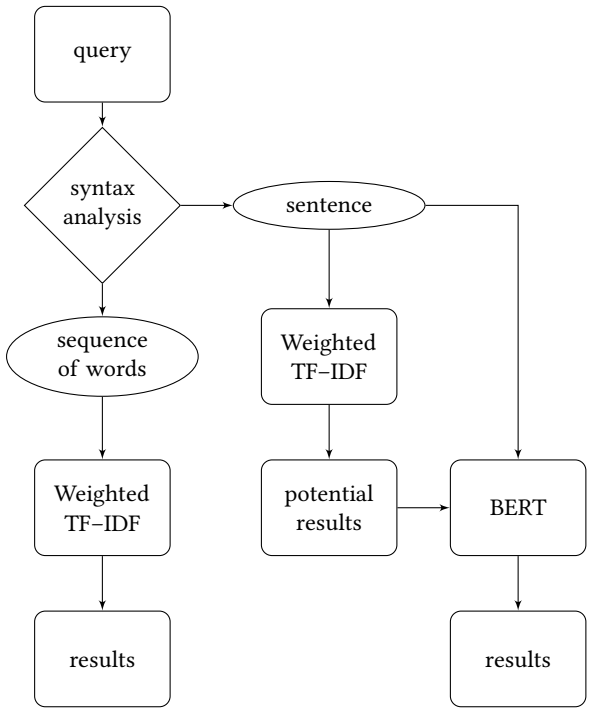


Figure 1: General approach

The Weighted TF–IDF model, as discribed in section 4.5, gives higher weight to the terms in questionable spans. Thus, we first need to find out the questionable spans from the target set of text data. We use a BiLSTM-CRF model (see section 4.4) to do this. We train the model with a reading comprehension dataset, SQuAD

(Stanford Question Answering Dataset, see section 4.3), and use the model to tag all the questionable spans in the sentences.

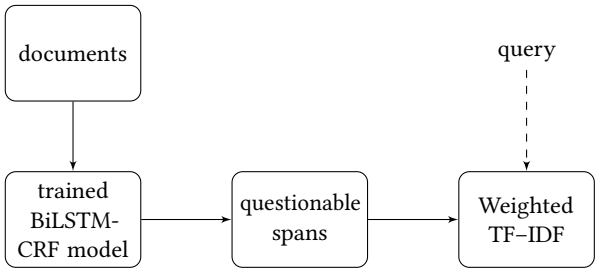


Figure 2: Weighted TF–IDF

4.2 Syntax Analysis Tree

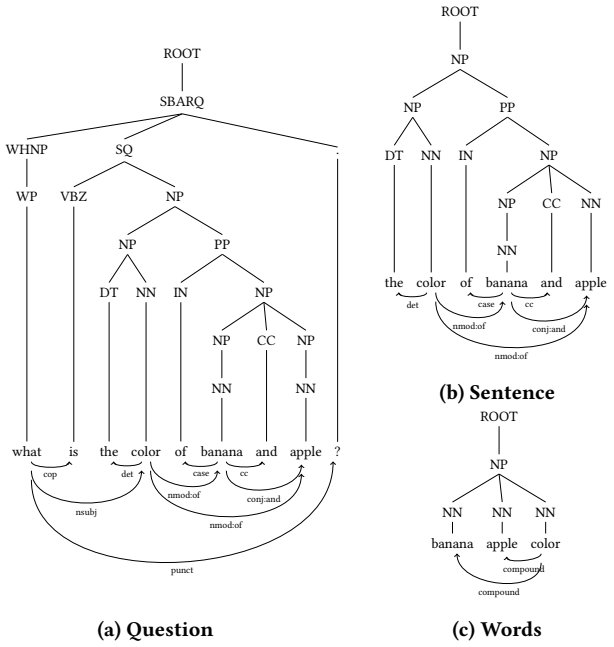
Syntax analysis is an NLP task to determine the structural role of words in the sentence. More specifically, syntax analysis will give each word a syntax label and the word's position in its syntax tree. There are two major syntax analysis types, one is the dependency parse and the other is the constituency parse. The goal of the dependency parse is to find the dependency relationships between "head" words and words that modify those heads. The goal of the constituency parse is to parse a sentence into different phrase structures.

Our task in this part is to determine different user inputs. We assume there are three different user input types: Question, Sentence, and Words. First, the "Question" type can be defined as an interrogative sentence. We can handle this kind of user input via Text Comprehension model, Weighted TF–IDF model, and question-suggestion model. Second, the "Sentence" type can be defined as all meaningful and grammatical sentences except interrogative sentences. We can handle this kind of user inputs via Text Comprehension model and Weighted TF–IDF model. Third, the "Words" type is the user input that is neither the "Question" type nor the "Sentence" type, which means it is just a sequence of words. Since it is unlikely for Text Comprehension model to generate some useful sentence representation for this kind of user input, we handle it with only Weighted TF–IDF model.

We notice that some labels in syntax analysis are very useful for this task. In this case, we simplify this task by only using two labels to determine the type of user input — "SBARQ" in constituency parse and "nsubj" in the dependency parse.

For an interrogative sentence, constituency parse provides us with a very useful label "SBARQ", which refers to a direct question introduced by a wh-word or a wh-phrase. For other grammatical sentences, we think that it should have a "nsubj" label in its dependency parse because a normal sentence should have a subject. If the user's input is only a sequence of words instead of a meaningful sentence, we will only get several "compound" labels. Here is a simple example of the parsing result of these three types of user inputs [**?**].

As for the implementation part, we use spaCy and one of its plugins "Berkeley Neural Parser" to get the Constituency Parse's and Dependencies Parse's results of the sentence.



(a) Question **(b) Sentence** **(c) Words**

Figure 3: Syntax Analysis Tree

4.3 SQuAD Dataset

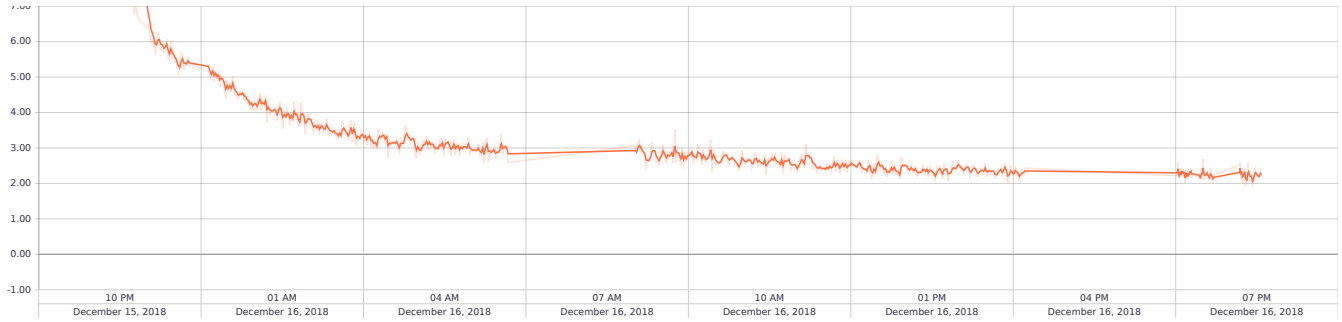
The Stanford Question Answering Dataset (SQuAD)[**?**] is designed for reading-understanding tasks. However, by utilizing this dataset, our goal is to find the important parts of given sentences and increase the corresponding weights when applying TF–IDF. In this dataset, the answer field comes from the original text, so we can take the answers in the original text as questionable and therefore, important parts. Furthermore, answer fields are represented as spans in the original text, so we can convert those spans into label sequences. Finally, we can use these two sequences (original text sequence and label sequence) to train the BiLSTM-CRF Model for sequence labeling tasks.

```
Born and raised in Houston, Texas, she performed in
    ↪ various singing and dancing competitions as a
    ↪  child, and rose to fame in the late 1990s as
    ↪  lead singer of R&B girl-group Destiny's
    ↪ Child.
[0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0,
    ↪  0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 0,
    ↪  0, 0, 0, 1, 1, 1, 0]
```

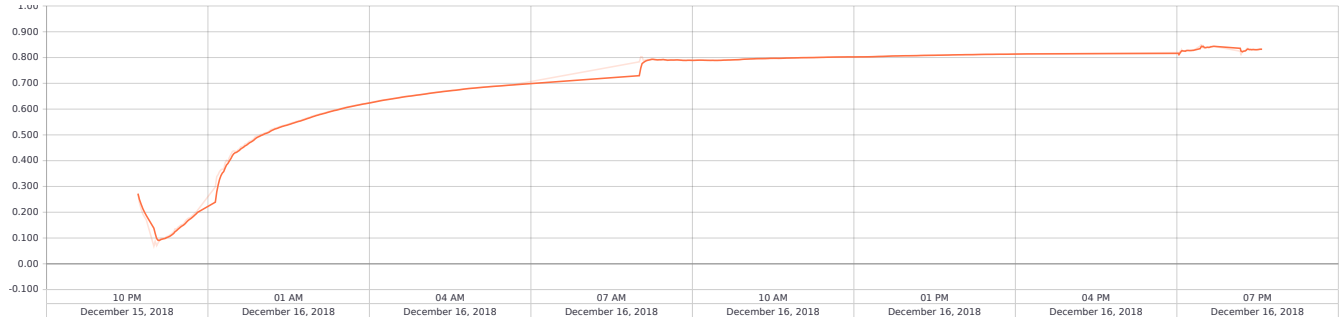
Figure 4: Caption

4.4 BiLSTM-CRF Model

The task of finding out the "questionable parts" of given documents is similar to the sequence labeling task. Our task is learning a pattern to mark tokens that belong to questionable part as 1 and mark others as 0. **?** 's work [**?**] inspires us to implement this part.



(a) Loss



(b) F1

Figure 5: Loss and F1

BiLSTM is a bidirectional version of LSTM, which generates words' representation based on the information from both forward and backward directions. LSTM is a kind of units of a recurrent neural network (RNN). A common LSTM unit is composed of a cell, an input gate, an output gate, and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell [**?**]. We use LSTM model, instead of some traditional machine learning such as random forest or AdaBoost, to make the labeling model contextualized. This is important because our task is to get a "span" of words instead of several scattered words as questionable parts. A contextualized model is more likely to mark a continuous sequence as 1 or 0.

The next step is to use conditional random fields (CRFs) to mark the questionable parts. CRF is a kind of sequence model used to predict labels for a sample sequence with context information. Specifically, CRF is very useful in applications such as POS Tagging, named entity recognition, etc., being an alternative to the related hidden Markov models (HMMs). Those applications are related to our task of giving 0–1 labels to tokens. It is proved that combining CRFs and BiLSTM is one of the best and most efficient ways for sequence labeling. CRF works in a similar way as Hidden Markov Model (HMM), predicting the most possible label paths of a given sequence by calculating the global conditional transfer probabilities, so that our model takes global information into consideration.

Due to the limitation of computing resources and the amount of training data we have, this is not the final evaluation of our BiLSTM-CRF model. We look forward to better performance of this model.

4.5 Weighted TF–IDF

TF–IDF (term frequency–inverse document frequency) is a popular term weighting scheme intended to show the importance of a word to a document in a set of documents. Given a user query, accumulating TF–IDF values of each term in the query gets different scores for different documents, which shows the relevance between the query and the documents.

TF–IDF is the product of TF (term frequency) and IDF (inverse document frequency). TF simply shows the number of times a term occurs in a document, while IDF helps to diminishes the weight of discriminative terms like "the" hence to increase the weight of terms that occur rarely in the whole corpus. TF and IDF can be calculated as

$$\mathrm{tf}(t,d) = \mathrm{count}(t,d)\,,$$

$$\mathrm{idf}(t,D) = \log \frac{|D|}{1+|\{d \in D : t \in d\}|}\,,$$

where $\mathrm{count}(t,d)$ is the number of times term t occurs in document d, $|D|$ is the total number of documents in the corpus D, and $|\{d \in D : t \in d\}|$ is the number of documents where the term t appears. Thus, TF–IDF can be calculated as

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D)\,.$$

Although IDF catches the importance of a term in a corpus and diminishes influences from discriminative terms, it determines the amount of information a word contains based on a corpus-wide evaluation. With the intention to let our model pay even more attention to the document-wise more meaningful parts, i.e., questionable spans, we introduce Weighted TF–IDF.

Weighted TF–IDF adds a weight w to TF–IDFs of terms that are included in the questionable spans of a document, so that the terms containing more meaningful information in a document contribute more to its score. We define Weighted TF–IDF as

$$\text{w-tfidf}(t,d,D) = \begin{cases} \text{tfidf}(t,d,D), & \text{if } t \notin \text{qs}_d, \\ w \cdot \text{tfidf}(t,d,D), & \text{if } t \in \text{qs}_d, \end{cases}$$

where qs_d is the questionable spans of document d.

Implementing the TF–IDF model, we create inverted index mapping terms in the corpus to the documents containing them in order to make a speedy search. We also create another inverted index mapping terms to the documents containing them as a token in the questionable spans, so that we can check if a TF–IDF needs to get w times weighted more efficiently.

4.6 BERT Model

BERT (Bidirectional Encoder Representation from Transformers) is a state-of-the-art pre-training language representations with which we can generate the sentence embedding (like word embedding). Like ELMo[**?**], BERT also uses multilayer network to capture the text meaning. However, BERT introduces self-attention mechanism and Transformer[**?**] to encode input token, making it the best language representations model nowadays.

We use the last layer of BERT hidden layer and sum up each token's output vector into one vector used to represent sentence embedding. Notably, the token's output vectors here do not include the CLS token and the SEP token. The reason is that these two tokens play similar rols and create similar vectors even in different token sequences, which could decrease the distinction of different sentences and make it difficult for ranking.

With the sentence embedding created by BERT, we are able to calculate the similarity of different sentence via creating sentence embedding. We call this kind of similarity as "semantic score", which could be used to introduce semantic information in our retrieval rank task. Because we already are able to get the weight TF–IDF score as the previous section described, the specific approach is to add this two scores together and make a final sort. It is reasonable because the information those two scores represented is orthogonal. The TF–IDF score represents the scenario that the user input exactly match the documents; the semantic score represents the scenario that user want to search some relevant documents. Even BERT is the current best model for language representations, it still is a black box model and very sensitive to noise. Moreover, a more common scenario is that user expect search result exactly matches user's input, and semantic information only is a supplement for traditional keywords searching approach. On the other hand, TF–IDF model does not take semantic information into consideration at all. It is a compromise approach to combining the two score together. If users' input is a meaningless token sequence, we should only consider the Weighted TF–IDF score.

In order to reduce the computation on similarities for all sentence pairs, we first use Weighted TF–IDF to rank documents and get a larger number of top documents than needed, say 100, as potential results. Then we use BERT to calculate the similarities between user's query and these 100 candidates, re-rank the documents and get the top 10. This helps to get results more accurately and efficiently when the corpus is very large and there are many of them getting similar TF–IDF scores. For example, if a user search with a query "red apple" from a corpus where there are two documents, "the red apple is placed in the house" and "the apple is placed inside the red house", TF–IDF gives the same score regardless of different semantic information. Here is where BERT comes in. With BERT, our goal is to disambiguate the top-ranked TF–IDF results and make give them a better ranking.

5 EXPERIMENT

5.1 Setup

To illustrate the effectiveness of our approach in improving the precision of semantic search, we would like to set up an experiment. We used the Yahoo! Answers Manner Questions v2.0 dataset as our experimental dataset, which is a subset of the Yahoo! Answers corpus from a Oct 25, 2007 dump. It is a small subset of the questions, selected for their linguistic properties. Additionally, questions and answers of obviously low quality are removed from the dataset, i.e. only questions and answers that have at least four words, out of which at least one is a noun and at least one is a verb, are kept.

We followed two steps to generate a preprocessed database. First, we used the BiLSTM-CRF model trained with the SQuAD dataset to identify all the questionable spans in the answer set of our experimental dataset. This is for the purpose of calculating the Weighted TF–IDF score later. Second, we used the BERT natural language model to convert all the sentences in the answer set into high-dimensional vectors based on their semantic distance.

We then randomly selected 1000 questions from the question set of the experiment dataset as user input queries. We used three models to compute the top 5 most relevant ranked sentences individually. These three models were the traditional TF–IDF method, the Weighted TF–IDF method and our proposed approach, the Weighted TF–IDF method combined with the BERT model.

We hid the source models of the output and let human readers to mark the relevance of the output based on the corresponding input question. With these labeled relevances, we finally used three metrics to compare the experiment results — P@3 (Precision at 3), P@5 (Precision at 5) and NDCG (Normalized Discounted Cumulative Gain). Precision at a particular rank position k is defined as

$$\text{P@}k = \frac{|\{\text{relevant docs}\} \cap \{\text{retrieved docs}\}|}{|\{\text{retrieved docs}\}|},$$

and NDCG at position p is defined as

$$\text{NDCG}_p = \frac{\text{DCG}_p}{\text{IDCG}_p},$$

where $\text{DCG}_p = \sum_{i=1}^{p} \frac{rel_i}{\log_2(i+1)}$, and $\text{IDCG}_p = \sum_{i=1}^{|\text{REL}|} \frac{rel_i}{\log_2(i+1)}$.

5.2 Result

We provide the experiment results of the three metrics mentioned above in Table 1. The numbers are the mean value computed based on the outputs of 1000 input queries using the corresponding experiment setup.

Figure 7 is an example of the outputs given by two different models with the same input query. The first 5 sentences are generated

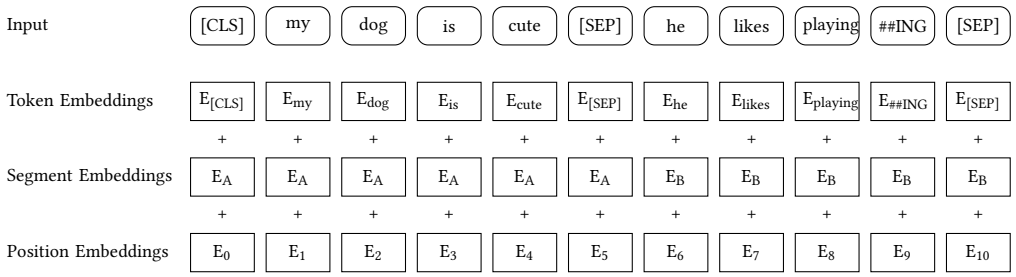


Figure 6: Bert Tokens

```
Input anything >> What is Yale famous for
Searching "what is yale famou.." with WEIGHTED TF-IDF + BERT...
  1. The Boston Globe wrote that "if there's one school that can lay claim to educating the nation's top
       ↪ national leaders over the past three decades, it's Yale."
  2. When challenged on the distinction between Dukakis's Harvard connection and his own Yale background, he
       ↪ said that, unlike Harvard, Yale's reputation was "so diffuse, there isn't a symbol, I don't think, in
       ↪ the Yale situation, any symbolism in it" and said Yale did not share Harvard's reputation for "
       ↪ liberalism and elitism".
  3. Yale is noted for its largely Collegiate Gothic campus as well as for several iconic modern buildings
       ↪ commonly discussed in architectural history survey courses: Louis Kahn's Yale Art Gallery and Center
       ↪ for British Art, Eero Saarinen's Ingalls Rink and Ezra Stiles and Morse Colleges, and Paul Rudolph's
       ↪ Art & Architecture Building.
  4. His father is a black Kenyan," in a column entitled "What Obama Isn't: Black Like Me."
  5. What he means is he dislikes having his metaphysics criticized."

Input anything >> !to
You have switched to PLAIN TF-IDF ONLY.

Input anything >> What is Yale famous for
Searching "what is yale famou.." with PLAIN TF-IDF...
  1. Some public rituals could be conducted only by women, and women formed what is perhaps Rome's most famous
       ↪ priesthood, the state-supported Vestals, who tended Rome's sacred hearth for centuries, until
       ↪ disbanded under Christian domination.
  2. The famous tirade of Lucretius, the Epicurean rationalist, against what is usually translated as "
       ↪ superstition" was in fact aimed at excessive religio.
  3. His most famous remark on religion is that "religion is what the individual does with his own solitariness
       ↪ ... and if you are never solitary, you are never religious."
  4. Between 10 and 17 October 1757, a Hungarian general, Count áAndrs Hadik, serving in the Austrian army,
       ↪ executed what may be the most famous hussar action in history.
  5. The Reverend Ezra Stiles, president of the College from 1778 to 1795, brought with him his interest in the
       ↪ Hebrew language as a vehicle for studying ancient Biblical texts in their original language as was
       ↪ common in other schools , requiring all freshmen to study Hebrew in contrast to Harvard, where only
       ↪ upperclassmen were required to study the language and is responsible for the Hebrew phrase Urim and
       ↪ Thummim on the Yale seal.
```

Figure 7: An example of outputs

by our approach (Weighted TF–IDF + BERT). The other 5 sentences are generated by the traditional TF–IDF method.

5.3 Evaluation

As reflected in the Table 1, our proposed Weighted TF–IDF method outperforms the traditional TF–IDF method in all three metrics. This shows that by identifying the questionable spans in the text and give them higher weights when calculating the TF–IDF score, we are more capable of finding the most relevant answers to user queries.

Moreover, the BERT natural language model aids the Weighted TF–IDF method in terms of providing semantic relevance measures

Table 1: Experiment Result

	P@3	P@5	$NDCG_5$
TF–IDF	0.243	0.214	0.366
Weighted TF–IDF	0.293	0.272	0.412
Weighted TF–IDF + BERT	0.529	0.501	0.519

so as to better compare the true semantics between the user query and potential answers. As we can see in Table 1, the model of the Weighted TF–IDF method together with the BERT model outperforms the traditional TF–IDF method by a significant margin in all three metrics.

We provide a sample output generated by these two models in Figure 7. There are 3 sentences in the results of our proposed model that are considered semantically relevant to the user query. In contrast, there is only 1 sentence in the results of the traditional TF–IDF method that is perceived as semantically relevant to the user input. This well illustrates that our proposed model indeed improves the semantic search precision as compared to the traditional TF–IDF method.

However, the precision and cumulative gain we get from both our proposed approach and the traditional TF–IDF method are still quite low. We believe that it is caused by the diverse nature and relatively small size of the experiment dataset. There may not be enough semantically similar documents in the dataset, which means that even a perfect semantic search engine cannot find 5 relevant answers to all the user input queries. Nonetheless, our proposed approach still outperforms existing methods by a significant margin and is able to rank the results properly according to their relevance.

6 CONCLUSION

In this paper, we defined a new concept of "Questionable Spans" which refers to the parts in a text that stands a high chance of forming the answer to a question targeting the original text. Given that, we proposed a novel approach which identifies the questionable spans in a sentence and gives higher weights to them when calculating the TF–IDF score. We then incorporate the BERT natural language model to complement the TF–IDF framework in terms of providing true semantic relevance measures of potential answers during a semantic search. Based on our experiment results, our proposed approach outperforms existing methods like the traditional TF–IDF method by a significant margin in both the precision and cumulative gain of the search results.

We plan to extend our proposed approach in several ways in the future: bringing in more features such as the depth of the word in the syntax tree when training the BiLSTM-CRF model, and extending the BERT model to handle the case when users input a sequence of words instead of a complete sentence.