# Evidence Selection in Long Documents

**Subendhu Rongali**
CICS, UMass Amherst
srongali@cs.umass.edu

**Rajarshi Das**
CICS, UMass Amherst
rajarshi@cs.umass.edu

## 1 Introduction

A lot of Natural Language Processing tasks currently consist of processing a few sentences or paragraphs of text to solve a certain downstream task. Question-Answering in the SQuAD dataset consists of a single question sentence and a paragraph of context. Tasks like named entity recognition and semantic parsing also mostly contains datasets where the inputs are fairly small.

In most real world applications however, one would have to go through entire documents to be able to solve a problem. Take the problem of comparing two treatments for a disease that are mentioned in a clinical trial report. One would have to browse through the entire document to find the relevant text to be able to answer the question of which treatment is better. This problem of finding the evidence in large documents for downstream tasks exists in other scenarios as well. Going back to the bio-medical domain, one would have to search through entire clinician reports to find evidence of drug interactions.

Researchers have released datasets and have started working on methods to tackle these tasks consisting of long documents. In this project, we aim to tackle one such problem, the task of the treatment comparison.

Our task is defined as follows: Given a PubMed article about a randomized control trial, answer a structured question about treatments in the trial. The structured question is a prompt that contains an intervention, comparator, and an outcome. The intervention and comparator are some treatments that are described in the article. The outcome is a patient outcome that is of interest. The answer to the question is one of the three options: significantly increase, significantly decrease, and no significant difference. The task is described with an example in Figure 1. In the example, we

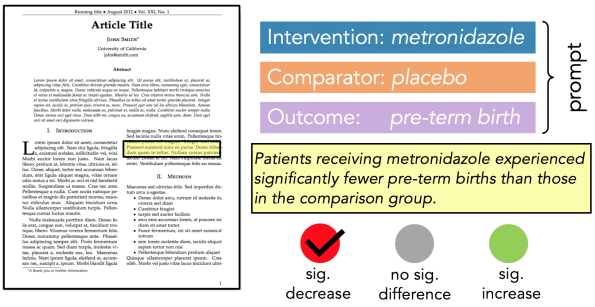


Figure 1: Our task: Evidence selection to answer questions about different treatment options, structured as a prompt. We split this task into two steps, the evidence selection part, and the final prediction.

compare the drug metronidazole (intervention) to placebo (comparator) for their effect on pre-term birth (outcome). It is clear to see from the given evidence text that the answer to the prompt would be significantly decreased.

Researchers (Jia et al., 2019; Lehman et al., 2019) have solved this task of finding evidence in documents by training models to softly attend to relevant sentences for their task. This attention is however soft and was shown to provide only a small fraction of the attentional mass to the actual evidence tokens (Lehman et al., 2019). The real evidence is as we know, probably a few sentences which calls for a more peaked or hard attention distribution.

Moreover, due to size of the text in these tasks, state-of-the-art pretrained language models like BERT and ELMo have not been used in these tasks. It is practically impossible to fine-tune a massive language model like BERT on a document level task in an end-to-end fashion.

In this project, we plan to explore and exploit these two limitations. We propose a pipe-lined architecture that gets away from the soft attention distributions by hard selecting evidence sentences.

Next, we train a powerful classifier built on SciBERT to make predictions based on the selected evidence.

We obtained good results for the second part of our pipeline. Including a BERT model helped the predictor based on oracle evidence to perform really well. We obtained an F1-score of 81.6 beating the score reported in Lehman et al. (2019) by around 8 points. The first part of our pipeline, the evidence classifier however performs poorly. We obtain a classification accuracy of 84.2% but including it in the overall pipeline only gives us an F1-score of 42.2, which falls behind the previous scores. We discuss our findings and possible reasons for this and end with proposals for how to improve this part of the pipeline to obtain better results.

## 2 Proposed Approach

In this section, we describe the pipelined approach we experimented with to solve the given task.

Going forward, we pick SciBERT (Beltagy et al., 2019) as our choice of language model for this task since it has been trained on lots of open domain bio-medical data.

We evaluate two methods for the evidence classifier in the pipeline architecture. First, we use a simple BM25 algorithm to predict which sentences are evidence and which are not. Second, we train an evidence classifier augmented with SciBERT to make these predictions.

We fix the second part of the pipeline to a SciBERT classification model that simply makes a multi-class prediction. We call this the Oracle Predictor since it makes predictions assuming it is given the correct evidence.

### 2.1 The BM-25 Classifier

We choose the BM-25 implementation from the bm25 python library. Our query is constructed by concatenating words from the intervention, the comparator, and the outcome, and the words 'significant', and 'difference'. We then use a sliding window to go through the sentences in the given document, and obtain the score from bm25 for each of them with the constructed query. We pick the best sliding window of sentences and use this as the evidence for the next step. This is shown visually with in example in Figure 2.

### 2.2 The SciBERT Evidence Classifier

The total number of evidence sentences in a given article is a very small number compared to the total number of sentences. So if we simply train a classifier to predict these labels, we face a huge issues of class imbalance and other inefficiencies.

So for this work, we draw ideas from literature in word embeddings to generate positive and negative examples for the evidence classifier. For a positive sample, we simply select a sentence that is marked as evidence. For a negative sample, we have a more involved approach.

First, we create three categories of negative example sentences given a prompt.

- Sentences from the article that are marked as non-evidence.
- Sentences that are marked as evidence for prompts in other articles.
- Sentences that are marked as evidence for other prompts in the same article.

The idea behind this is to generate difficult training examples for the classifier by using other possible evidence sentences. We can see that the model would face the hardest challenge from the third category of negative examples. We sample negative examples from each category for each positive example.

We include the intervention, comparator, and the outcome in this decision by simply adding them to the document sentences, along with a BERT separator token. This is to ensure that the evidence classification is informed by the given prompt.

Now, the model can be trained with two different kinds of losses. Since we generate positive-negative pairs through sampling, we can optimize the max-margin, which corresponds to a hinge loss formulation. We can also simply use BCE loss to train the model. We experiment with both the settings in our experiments. This process is visually shown in Figure 3 with an example.

### 2.3 The Oracle Predictor

The oracle predictor is a simple SciBERT classifier. We train the model by creating inputs that include the prompt and the evidence to perform a 3-class classification task. The input is generated by concatenation the intervention, the comparator, and the outcome, followed by the separator token,

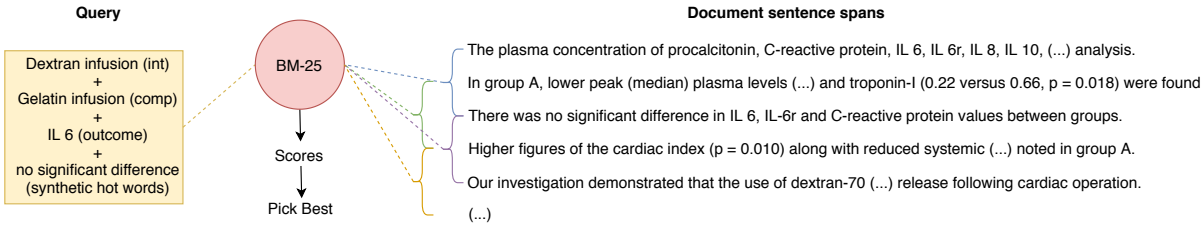


Figure 2: The use of BM-25 to select the best evidence span. Thw window size shown is 3 with a stride of 1.

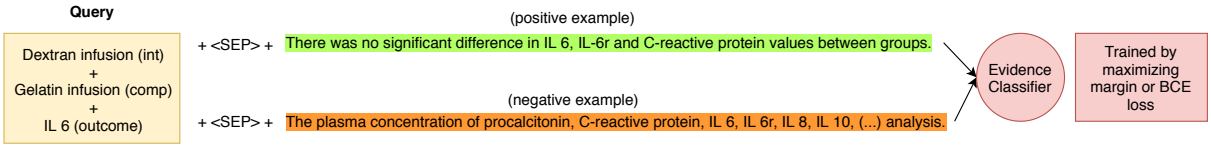


Figure 3: The training process for the evidence classifier with SciBERT. The negative example shown here is from category 3 i.e a random non-evidence sentence from the same article.

and then the article, in a similar fashion to the evidence classifier.

## 3 Dataset

The dataset used in this work is from Lehman et al. (2019). It consists of various prompts for treatments in 2419 PubMed articles. Each article has one to many prompts that were created and verified by physicians. The prompts contain the intervention, comparator, outcome, and the label, along with the evidence sentences for the article. The dataset is split into train, test, and validation sets and we follow the same splits for our experiments. There are 8147 examples in train, 964 in test, and 1002 in the validation sets.

## 4 Experimental Setup

For the BM-25 classifier, we use a stride of 1 and a sentence window size of 3 to extract candidate evidence spans.

For the SciBERT trained evidence classifier, we sample positive and negative samples as described and randomly append a few sentences of context around the sentence. We make sure that the number of sentences is again 3 here.

All models were built using the allennlp pytorch library. SciBERT weights were downloaded from the allennlp homepage. The models were trained with the recommended settings for BERT models: the Bert-Adam optimizer with learning rate $2e-5$ and 10% of the data as warm-up steps. All the other setting were set to defaults in the allennlp implementation.

| Model | F-score |
|---|---|
| Heuristics (Lehman et al., 2019) | 45.3 |
| Logistic Regression (Lehman et al., 2019) | 73.1 |
| Neural Network (Lehman et al., 2019) | 73.9 |
| ELMO classifier | 77.6 |
| BERT classifier | 78.7 |
| **SciBERT classifier** | **81.6** |

Table 1: The results of various models in the oracle prediction setting. Our best model outperforms previous approached by almost 8%.

## 5 Results and Discussion

### 5.1 The Oracle Predictor

We found that our oracle predictor receives a huge boost in performance due to the addition of SciBERT. Our model does almost 8% better than the models reported in Lehman et al. (2019) for the oracle setup i.e predicting the label given the true evidence.

We also experimented with the traditional BERT and ELMo embeddings instead of SciBERT and found that SciBERT achieves the best performance. All our scores are reported in Table **??**.

### 5.2 The Evidence Classifier

This section doesn't have prior results since we created it as a sub-task in our pipeline. We hence report only the results of our models.

Our evidence classifier was trained with 50% positive and 50% negative evidence sentence spans. We initially achieved promising results for the classification task. Training our architecture with either the Hinge or BCE losses achieved ac-

| Model | Accuracy |
|---|---|
| Classifier with Hinge Loss (m=0.7) | 79.3 |
| **Classifier with Hinge Loss (m=0.5)** | **84.5** |
| Classifier with BCE Loss | 81.2 |

Table 2: The classification performance of various evidence classification models we trained.

| Model | F-score |
|---|---|
| Heuristics (Lehman et al., 2019) | 35.4 |
| Logistic Regression (Lehman et al., 2019) | 42.3 |
| NN + attention (Lehman et al., 2019) | 50.5 |
| **NN + pretrained conditional attention** | **53.1** |
| BM-25 + SciBERT Predictor Pipeline | 50.3 |
| SciBERT Classifer + Predictor Pipeline | 42.2 |

Table 3: The results of our pipeline models. The BM-25 algorithm performs decently despite its simplicity but out SciBERT evidence classification models fail.

curacies of over 80% on a held out validation set. The results are shown in Table 2.

For the Hinge loss, a hinge margin of 0.5 gave us the best results. With larger margins ($> 0.8$), we believe we achieved worse results since the model is trained with a stricter constraint and it over-fits to some data points. We also found that the model couldn't learn anything with a high margin of 0.9.

For these experiments, as described in the experimental setup, we also added a few random sentences around each sentence to provide more context in classification. We found that simply using each sentence is not enough. To provide a concrete example, the sentence "Group A patients fared better than those in Group B here." is a sentence that is marked as evidence. However, the descriptions of groups A and B, which contain the information of possible interventions and comparators, and the overall outcome are not provided here. They are sometimes presented a few sentences before, whereas in other cases, they are completely in a different part of the article, presenting us with a huge challenge of coreference resolution. To be able to handle cases where this information is presented a sentence or two ago, we made the choice of adding the context sentence of a fixed size.

Despite our initial results, we found that our classifiers that were trained in this fashion were not very useful in the overall pipeline. We describe these results next and provide some observations and future improvements.

### 5.3 Overall Pipeline

As described earlier, we experimented with two overall pipeline architectures. The first is the BM-25 to pick the top evidence span, followed by the SciBERT oracle predictor. The second is the SciBERT classifier, followed by the same predictor.

Our initial results showed that we had a decent classifier, and a really well performing oracle predictor. Putting these together in the overall pipeline, we expected to see a good boost in performance. However, we found that the trained classifier was not actually good in the overall pipeline. The result reported here is for our best classification model, the model trained with a hinge margin of 0.5. The BM-25 pipeline model showed promise however.

The results of our models and the previous approaches in Lehman et al. (2019) are shown in Table 3. Note that the previous NN models are both end to end. This is possible with the huge article sized due to the simplicity of the architecture (no BERT etc).

Our classifier could be doing worse due to a number of reasons. First, the distribution of positive and negative evidence sentences is different during the classifer training and the overall pipeline. We use a 50-50 split to train the classifier while the true positive evidence is actually a very small fraction of the overall sentences. We initially believed this wouldn't affect our models too much since we perform a ranking and choose the best evidence, and not a true classification task in the pipeline. However, this doesn't seem to be the case. Second, we refer back to something we already mentioned about the context of each sentence. We found that in many cases, the intervention and comparator are not actually a part of the evidence sentence but are rather described in a previous paragraph of section. This provides a huge challenge for any classifier since it simply doesn't have enough context to make the decision. We including a few sentences of context to mitigate this issue but it appears that there is still a long way to go.

We conclude by providing some ways to improve our models. First, we could change the distribution of the data while training the evidence classifier to be more skewed towards negatives. We could also perform a post-processing step for

the evidence attention scores over the sentences to accumulate and peak the scores for sentences across different spans before sending them to the predictor. Finally, we could try to train an end-to-end model using some smaller and newer variants of BERT such as DistilBERT (Sanh et al., 2019), ALBERT (Lan et al., 2019) etc. We could also weakly link the evidence classifer and predictor in our pipeline by propagating some information back to the classifier during the prediction steps.

## 6 Related Work

Jia et al. (2019) propose a document level relation extraction models that works by extracting and aggregating contextual representations of mentions and mention pairs across the whole document. This architecture is built over a simple bi-LSTM encoder.

The evidence selection for treatment comparison task was introduced by Lehman et al.. The models in this paper use soft-attention over words of the entire documents and use a bi-LSTM encoder. We plan to switch both of these components with the ones we proposed earlier.

Further, there has been work in open domain question answering across multiple passages which can be seen to be of the length of a document when combined (Wang et al., 2019). The authors propose a model applied BERT to individual passages and then normalizes scores across all the passages to obtain better answer predictions.

The idea of using a retriever to find relevant sentences from documents and then using those extracted sentences was applied on a document-corpus level by Das et al.. We plan to use a similar mechanism for our document level task.

There has also been work that looks at applying capsule networks to solve these kinds of NLP tasks (Zhao et al., 2019). The major takeaway from the papers is the optimized routing process that enable these networks to scale to challenging NLP applications like ours. These networks however are still in a nascent stage and it is not clear how they can be integrated and trained with language models.

## 7 Conclusions and Future Work

In this work, we propose and evaluate various pipeline models to solve the task of evidence inference for treatment comparison. Our approach of splitting the task into evidence classification, and prediction, couldn't beat existing end to end models despite achieving good results in the individual stages. We presented all our results, analyzed shortcomings of our models, and proposed new directions to improve these models.

## References


Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. Scibert: Pretrained contextualized embeddings for scientific text. *arXiv preprint arXiv:1903.10676*.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step retriever-reader interaction for scalable open-domain question answering. *arXiv preprint arXiv:1905.05733*.

Robin Jia, Cliff Wong, and Hoifung Poon. 2019. Document-level $n$-ary relation extraction with multiscale representation learning. *arXiv preprint arXiv:1904.02347*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Eric Lehman, Jay DeYoung, Regina Barzilay, and Byron C Wallace. 2019. Inferring which medical treatments work from reports of clinical trials. *arXiv preprint arXiv:1904.01606*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage bert: A globally normalized bert model for open-domain question answering. *arXiv preprint arXiv:1908.08167*.

Wei Zhao, Haiyun Peng, Steffen Eger, Erik Cambria, and Min Yang. 2019. Towards scalable and reliable capsule networks for challenging nlp applications. *arXiv preprint arXiv:1906.02829*.