Team GESIS Cologne: An all in all sentence-based approach for FEVER

Wolfgang Otto
GESIS – Leibniz-Institute for the Social Sciences in Cologne
Unter Sachsenhausen 6-8
50667 Cologne
wolfgang.otto@gesis.org

Abstract

In this system description of our pipeline to participate at the Fever Shared Task, we describe our sentence-based approach. Throughout all steps of our pipeline, we regarded single sentences as our processing unit. In our IR-Component, we searched in the set of all possible Wikipedia introduction sentences without limiting sentences to a fixed number of relevant documents. In the entailment module, we judged every sentence separately and combined the result of the classifier for the top 5 sentences with the help of an ensemble classifier to make a judgment whether the truth of a statement can be derived from the given claim.

1 Introduction

Our approach is strongly related to the baseline approach. It is using a sentence retrieval method without more in-depth analysis of semantic properties of Wikipedia sentences as a first component. For the IR task no external resources beside the given Wikipedia sentences have been taken into account. The second component is the entailment model which is the same Decomposable Attention Model as the one used to generate the best baseline results in (Thorne et al., 2018). But in both components, there are some differences. In the IR component, there is no document retriever as a first step. Given a claim, we search directly on all Wikipedia sentences of the reference corpus for possible candidate sentences. In the entailment component, the difference lies not at all in the model, but in the data used during training and inference time. We trained the model sentence-wise and not claim-wise. I.e. we split the result set for each claim into combinations of claim and each sentence separately. To be able to handle more than one sentence evidence we introduce new classes. One class to identify evidence sentences which are part of supporting evidence with multiple sentences, and a second one to identify evidence sentences which are part of refuting evidence with more than one sentence.

2 Sentence Retriever

The basic idea of our evidence retrieval engine is the intuition that a preselection of specific Wikipedia articles for a given claim will exclude sentences, which are highly related to the claim-based on a word or even entity overlap, but will be excluded because the Wikipedia article has another topic.

Our approach tries to find sentence candidates from all sentences of the given shared task Wikipedia introduction sentences corpus. To keep the system simple and rely on a well-tested environment we indexed all sentences with a SOLR search engine[1] with the default configuration. Our idea to find relevant candidate sentences, which support or refute the given claim, is to identify those which are connected to the same entities or noun chunks. So we extracted those information from the claim and create a SOLR query to get a ranked sentence list from the search engine.

2.1 Preprocessing

Wikipedia Article Introductions: The problem of only working with single sentences is, that sentences of a Wikipedia article introduction loose the connection to the article title in many cases. To create good retrieval results we need a preprocessing step for coreference resolution to match sentences like “He is the biggest planet in our solar system.” from the article about Jupiter to the claim “Jupiter is larger than any other planet in the solar system.” We decided on the most straightforward solution and concatenated a cleaned version of the title to the Wikipedia sentences before indexing. For cleaning the title we cut of all parts beginning with a round bracket.

[1] http://lucene.apache.org/solr/.

Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), pages 145–149
Brussels, Belgium, November 1, 2018.

Also underscores will be replaced with spaces. So *"Beethoven_-LRB-TV_series-RRB-"*[2] will be transformed to *"Beethoven"* for example.

Query claim: For the generation of a query for a given claim we extracted all noun chunks and named entities with the natural language processing tool SpaCy[3] (Honnibal and Johnson, 2015) with the provided *en_core_web_sm* language model[4]. Then we filter all resulting individual words and phrases. Given this set of all words and multi-word units, we create a SOLR-query which is giving an advantage to adjacent words of the multi-word units which occur with a maximum word distance of two. Additionally, we query each word of each item in the set separately with a should query. The named entity *Serena Williams* for example is searched with a query where "Serena", "Williams" and "Serena Williams"~2 should all be matched. The swung dash in the last part of this query indicates that search results, where "Serena" and "Williams" occur with a maximum distance of two, will be pushed. The distance of two is chosen because it helps in cases like "Arnold Alois Schwarzenegger" to push the match of the search query "Arnold Schwarzenegger"~2. Here a more complete example:

> *Claim:*
> Serena Williams likes to eat out in a small restaurant in Las Vegas.
> *Named Entities:*
> Serena Williams, Las Vegas.
> *Noun Chunks:*
> Serena Williams, a small restaurant, Las Vegas
> *Unigram searchterms:*
> Serena, Williams, a, small, restaurant, Las, Vegas
> *Pushed bigram searchterms:*
> Serena Williams, a small, small restaurant, Las Vegas

The result of the sentence retriever is a list of sentences and their corresponding Wikipedia titles which matches best in concern of named entities and noun chunks based on the described extraction and querying.

3 Recognizing Textual Entailment

3.1 Preprocessing

During preprocessing of train and test data, we consider three steps. For the first step of tokenizing claim and Wikipedia sentence, we treat both of them differently. The Wikipedia sentences are already tokenized. The claims are tokenized with the standard SpaCy's rule-based tokenizer. For textual entailment, the same problem of coreference resolution described for the IR component pops up again. Because of this, we decided to add the title information to the Wikipedia sentences as additional information as well. Adding this information can help the entailment model identify the entity explained in the sentence. A working example:

> *Claim:*
> Stars vs. the Forces of Evil is a series.
> *Wikipedia title:*
> Star_vs._the_Forces_of_Evil
> *Wikipedia Sentence (Sentence No. 6):*
> On February 12 , 2015 , Disney renewed the series for a second season prior to its premiere on Disney XD .

Of course, it is a heuristic. There are sentences where the added information doubles the info of the entity. But then again there are sentences where the content does not match the entity described in the title. In practice, we join the tokens of the title to the sentence while excluding the additional information for disambiguation. I. e. for *"Hulk_(Film)"* we only add *"Hulk"* to the corresponding sentence string. For vectorization of the sequence token, we use GloVe word embeddings with a dimension of 300 produced with the method from (Pennington et al., 2014). To maximize the overlap to the words used in our Wikipedia-based dataset we used the ones trained on *Wikipedia 2014 + Gigaword 5* by the Stanford NLP group.[5]

3.2 Prediction Classes

The data set provides the special case where one single sentence is not enough to support or refute a claim. In this case, multiple sentence support

[2] Where *"-LRB-"* stands for *"Left Round Bracket"* and *"-RRB-"* for *"Right Round Bracket"* as it can be found in the already tokenized Wikipedia resources which were made available from the organizers of the competition.

[3] https://spacy.io/.

[4] https://spacy.io/models/en#en_core_web_sm.

[5] https://nlp.stanford.edu/projects/glove/.

is delivered. In 9.0% of the validation set claims where supporting or refuting evidence exists, a minimum of two Wikipedia sentences is needed. In 14.7% of the supporting/refuting claims, there is at least one possible multiple sentence evidence. Around 25% of the supporting or refuting sentences are part of multiple sentence evidence in the validation set. Multiple sentence evidence poses a problem for our approach of sentence-by-sentence entailment assessment. On the one hand, the class of a given claim and one sentence of multiple sentence support/refute cannot be classified as supporting or refuting. On the other hand, more information is delivered than in a regular *NOT_ENOUGH_INFO* claim sentence pair. We decided to deal with this by using not three, but five classes. For sentence-wise prediction we have extended the given classes *SUPPORTS*, *REFUTES* and *NOT_ENOUGH_INFO* with the two new classes *PART_OF_SUPPORTS* and *PART_OF_REFUTES*.

3.3 Generating NOT_ENOUGH_INFO sentences

In Thorne et al. (2018) the authors introduced two ways of selecting sentences for claims which are annotated as *NOT_ENOUGH_INFO*. They compared classifiers trained on randomly chosen sentences for this class with classifiers trained on data where the top 5 results of the sentence retriever are used as text input for them. The results show that on the textual entailment validation set both classifiers trained on random sentences show better results than the ones trained on top 5 results. But for the whole pipeline, the resulting accuracy drops around 1% for the Decomposable Attention Model (41.6% vs. 40.6% pipeline accuracy in Thorne et al. (2018)). As we used the same model, we decided to use the approach of selecting the top retrieved sentences for the *NOT_ENOUGH_INFO* annotated claims. To keep the number of sentences per class not too unequal, we chose to use the top 3 results of our sentence retriever for the test and validation set of the Decomposable Attention Model. It should be noted, however, that our sentence retriever is working in a slightly different way than the one used in the baseline approach.

For the occurrence of each label in the sentence-wise validation set for the entailment prediction task see table 1.

label	frequence
NOT ENOUGH INFO	19348
SUPPORTS	7012
REFUTES	7652
PART OF SUPPORTS	2741
PART OF REFUTES	2452

Table 1: Frequences of label in sentence-wise validation set.

3.4 Decomposable Attention Model

For the task of recognizing textual entailment we take a Decomposable Attention Model as described in (Parikh et al., 2016) and is one of the classifiers used in the baseline approach (Thorne et al., 2018). We selected the vanilla version of this network without self-attention on input sequences. This model compares each word vector representation of the input sequences with the representation of phrases of the other input sequence. The process of selecting words from the other sequence for comparison is called attention. After this, the representations for this comparisons are aggregated and in a final step used to predict, if one sequence supports or refutes the other or has not enough information for a decision.

The model is formulated with the aim of learning during training time which words are to be compared, how to compare them and in which entailment relation both sequences are to each other.

Basic Parameters: For training we used the given training and evaluation set for the shared task prepared and preprocessed as described above in a sentence-wise manner. We used batches of size 32 with equal number of words during training and a dropout rate of 0.2 for all three feed forward layers F, G and H of the neural network model. F, G and H are used here analogous to the terminology in (Parikh et al., 2016). We trained the model for 300 epochs on all batches of the training set and choose the best performing model measured on the validation set for the prediction of the test set. Tokens without word embedding representation in the *GloVe Wikipedia 2014 + Gigaword 5* (out of vocabulary words) are treated with the same approach as in (Parikh et al., 2016). The words are hashed to one of 100 random embeddings.

3.5 Ensemble Learner

The result of the entailment classifier is the judgment for each pair of claim and sentence of a Wikipedia introduction if the claim can be entailed from the sentence based on the five introduced classes. For taking part in the FEVER Shared Task, it is needed to decide on each claim one of the labels *SUPPORTS*, *REFUTES* or *NOT ENOUGH INFO*. The second part is to find the right sentence which underpins the judgment. The result of the sentence-based entailment classifier is a list of judgments which might be contradictory. As a result, we need a classifier which aggregates the results for the sentences to one final claim judgment. For this, we combine the entailment judgments from the classifier by using a random forest classifier (Breiman, 1999). As input, we take the probability of the judgments for all five classes of the top 5 results of the sentence retriever. In this way, the number of features can be summed up to 25. We kept the order of the sentences based on the sentence retriever results. We trained this aggregating classifier to predict one of the three classes awaited from the FEVER scorer for the evaluation of the shared task.[6] Together with the top 5 sentences of the sentence retriever, this result represents the output of the whole pipeline. To generate pipeline results for the validation set we trained the classifier on half of the validation set and predict the other half. For the FEVER Shared Task test set prediction we trained the classifier on all samples from the validation set.

4 Evaluation Results and Discussion

4.1 Sentence Retriever

To be able to measure the results returned from the retrieval component we take the FEVER scorer into account, too. To keep a different view on the outcomes, we measured the recall values for different allowed sizes of result sets. For additional analysis, we measured for each result set size separated recall values for only refuting and only supporting values. Figure 1 shows that recall values for the refuting sentences are lower than those for the supporting ones. This seems to reflect the intuition that in refuting sentences the word overlap to the claims are smaller than in supporting sentences. The fact that even with an allowed result set size of 100 the recall with our approach is

[6] https://github.com/sheffieldnlp/fever-scorer.

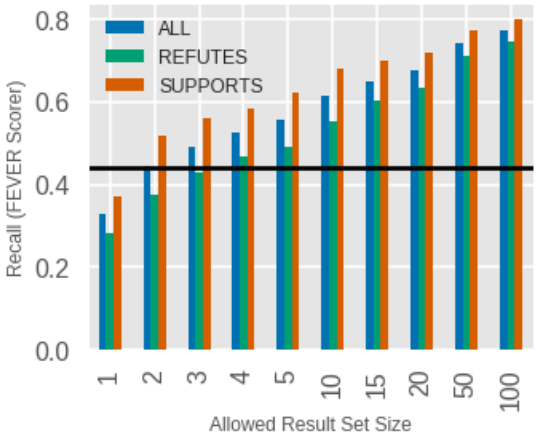

Figure 1: The recall values for sentence retrieval based on different allowed result set sizes. The black line shows the baseline recall for an allowed result set size of five[8]

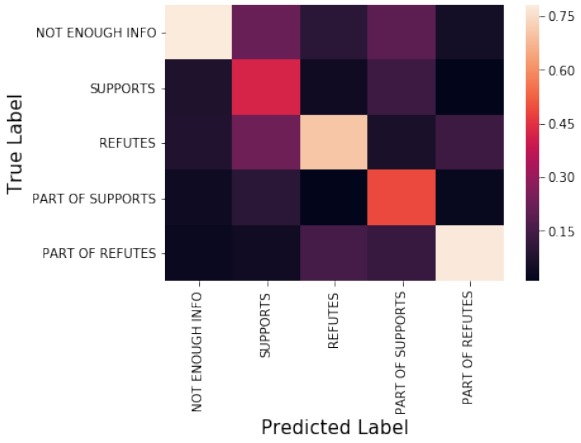

Figure 2: Heatmap of the precision of each class.

77.3% shows that a simple method which is dependent on word overlap between claim and sentences gets to its limits. In comparison to the baseline where 44.22% of the claims in the validation set were fully supported after the document and the sentence selection components, in our approach, 53.6% of them have full support, even though we do not use a document retrieval component at all. This would lead to an oracle accuracy of 69.1% (Baseline: 62.8%).

4.2 Entailment Classifier and full pipeline

The entailment classifier has an accuracy of 64.7% for the sentence-by-sentence prediction of the five classes. This is not comparable with the results of the baseline because of the sentence-wise comparison and the five label classification scheme. A

look at the class-wise precision of the classifier and the number of sentences per class draws attention to the fact that this value is strongly dependent on the number of sentences which are generated for the *NOT ENOUGH INFO* label. It is because for this class the model achieved the best precision values and the label is over-represented in the validation set.

As expected the classifier has problems to differentiate between refuting sentences and the ones, which are part of a multiple sentence refute. The same applies to the supporting sentences as you can be seen in Figure 2. For the all in all pipeline evaluation, we get a FEVER score of 46.7% on half of the validation set. The other half was used to train the aggregating ensemble learner. On the shared task test set, we achieved a FEVER score of 40.77 (8th place).

The next steps to evolve our system should be to focus on recall of sentence retrieving for refuting sentence and split up the strategies for both types of sentences.

References

Leo Breiman. 1999. Random forests - random features. *Technical Report 567*.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and verification. In *NAACL-HLT*.