## 1. Introduction

This reports aims to describe the system that we built for Fact Verification, which is a predominantly retrieval based system that can evaluate if a claim and can be validated by facts to either support or refute or there is not enough evidence found based on the information available in Wikipedia articles. The dataset used is derived from the FEVER[1] challenge, an ongoing competition to evaluate system's ability to verify information using evidence from Wikipedia (Thorne et al., 2018).

We utilized technologies including information retrieval, information extraction, and supervised deep learning to develop the Fact Verification Engine. The report also elaborates on the design choices that we took, the experiments done that led to the final approach and the performance of our system with its rank against a Codalab private competition held by the University of Melbourne.

The verification pipeline is divided into three main modules; relevant Documents Retrieval, evidence Sentence Selection and Claim Verification at a high level this is one of the most common pipelines as described in FeVer Shared Task (Thorne et al., 2018). In our implementation the former two are mainly IR based with re-ranking mechanism based on topic similarity and named entity extraction and the label identification is done via supervised deep learning.

## 2. Methodologies

We made three submissions to Codalab. The latest submission was beyond the baseline (20% in sentence selection and 40% in label accuracy). This section explains the methodologies that we use for submissions and further approaches that lead to even higher results according to the development set. A complete explanation of all the attempts that led to the final approach is in the experiments section.

| Submission | Label Accuracy | Document Selection F1 | Sentence Selection F1 |
|---|---|---|---|
| 1 | 35.5 | 55.5 | 25.6 |
| 2 | 38.1 | 54.7 | 39 |
| 3 (Final) | 40.7 | 56.3 | 39.5 |

### 2.1. Document Retrieval

We used Xapian[2] (an open source search Engine toolkit) as an initial step for identifying relevant documents. The provided corpus was indexed after applying standard preprocessing steps -tokenization, stemming, exclude stopword removal and NFD normalization- at the document level where in all sentences related to the same document are concatenated and indexed along with the document title. BM25 was used to score documents with relation to a claim query and document. BM25 ranking was used instead of TF-IDF as it is probabilistic based rather than similarity based. Our test as described in Experiments section has shown that BM25 had better performance in this context.

The top 50 documents relevant to the claim is extracted from the IR engine, then a re-ranking process is done based on cosine similarity of entities extracted from claim and document title, and also full text claim and document title similarity. This approach was inspired from team UNC-NLP and UCL Machine Reading Group, who were the top participants of the FEVER[1] challenge (Thorne et al. 2018). By investigating the training dataset, claim "*Nikolaj Coster-Waldau worked with the Fox Broadcasting Company*" have the first evidence document "*Nikolaj_Coster-Waldau*" appears as the first rank from IR. However, the second relevant evidence "*Fox_Broadcasting_Company*" appears in the 40th rank. After the re-ranking approach the evidence documents appear in the top 2 with similarity 0.57 for token matching and 0.69 for entities and capitalized words matching. Adding extra weight to the document title is intuitive as wikipedia topics usually contain the main fact the content is discussing.

Our first submission to Codalab was using a re-rank approach based on claim and document title token matching. Every document that has cosine similarity above 0.4 was taken. If no documents were found based on this threshold, the top 2 documents fetched from the IR engine. The threshold is set after intensive evaluation of the similarity between claim and document title in training set. Around 65% of sample records have cosine similarity based on title around 0.5, otherwise have similarity based on document sentences tokens around 0.2.

Since the approach gives more priority to title matching it does not cover evidences that has low similarity in title but high similarity in sentences. Thus, on the second approach we modified to combine the BM24 rank from IR engine and rank from entities, capitalized words, and tokens matching in document title. We return 50 documents per claim, take the top 2 based on each of IR engine rank, title matching, and entities and capitalized word matching. The document selection will be a unique set of this combination which returns at maximum six documents.

### 2.2. Sentence Selection

In the first submission we took all sentences from the selected documents that have token cosine similarity with claim above 0.2, if none was returned we take all sentences from the documents. The main drawback of this approach is there is an excessive number of sentences returned, thus, resulted in low precision relative to recall.

In the second submission we adjusted the document selection the provide at maximum six documents, then considered the cosine similarity between claim and each sentence from the selected documents. Each claim and sentence are transformed into embedded vectors using GloVe word2vec models. We sort sentences based on similarity score and filtered a maximum 5 sentences based on high threshold that gets reduced if no sentences where selected. For example, from all sentences we start to take only similarity above 0.95, if none is returned the threshold is reduced by 0.5, and repeated continuously until 0.6. This approach attempts to significantly increased sentence selection F1 on the test dataset from 25.6% to 39%, but with a

[1] http://fever.ai

[2] http://xapian.org

cost that the document selection F1 is decrease from 55.5% to 54.7%. If we relaxed the filtering, the precision declines and overall performance goes down.

### 2.3. Claim Verification

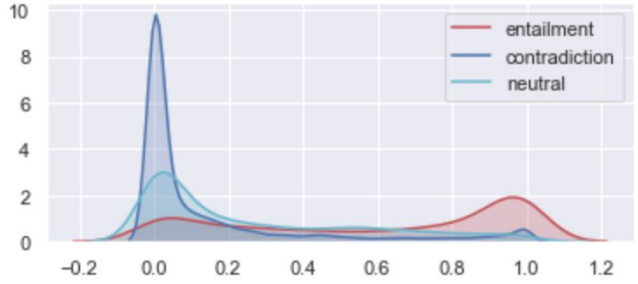
We initially built our training model using bag-of-words (BOW) model by merging claim and evidence sentences avalaible under either REFUTES or SUPPORTS label from training. For NOT ENOUGH INFO case as no documents we retrieved the first ranked document from IR. We used Neural Network Multi-Layer Perceptron (MLP) model with one hidden layer of 300 nodes using TF-IDF matrix from the BOW. The model was trained using 140,000 random records from training set.

As can be seen, the label accuracy on test data is below the baseline. During the experiments, we learned that the model is overfitting. The accuracy of the model using around 4,000 held-out dataset that is built on the same setting is around 87% but when it is implemented to development it decreases to 35%. The model is only applicable if we provide the relevant evidences as features. Since the sentence selection accuracy is low, the model cannot predict well.

To avoid overfitting, we introduced 'noise' for the next model. Instead of taking only evidence sentences, we train the model using all sentences from the evidence documents. This approach behaved as expected, using the same sentence selection from the first submission, the accuracy in development set went up from 32.95% to 34.23% and using the second approach for sentence selection it rose to 36.67%. The second codalab submission using this approach returns 38.1% label accuracy.

The main weaknesses of using BOW model is it cannot capture the semantic distribution within the sentences, both for claim and evidence. Hence, in the latest submission we used an ensemble technique by combining the result of second approach and a modified state-of-the-art pre-trained decomposable attention based textual entailment model using ELMo Embeddings by AllenNLP. The label prediction is set based on majority voting between the models and a weighted prediction to resolve a tie.

The textual entailment (TE) model produce three probabilities for every pair of sentences, entailment, contraction, and neutral. In this case, entailment can be associated with SUPPORTS, contraction with REFUTES, and neutral with NOT ENOUGH INFO. The challenge is to set the right threshold for the probabilities to predict the label.




We set the threshold based on the distribution of each probability in development data. It is noticeable that the entailment tend to have negative skew and the others have strictly positive skew. Therefore, we cannot set equal threshold for each class. Referring to the distribution, we set 0.98 on entailment probability as the minimum threshold to predict SUPPORT, 0.5 on contradiction to predict REFUTES, and otherwise predicted as NOT ENOUGH INFO.

Compared to the previous model, in development data both models predict more than 60% claims as NOT ENOUGH INFO, but SUPPORT prediction is higher in TE model and REFUTES in MLP model. Accordingly, if there is a tie between NOT ENOUGH INFO with the others, we take the others, and between SUPPORT and REFUTES, take SUPPORT from TE model, and REFUTES from MLP model. The following table shows the comparison in development data. The top one is from the MLP model and the below one is ensemble model as used for third submission. It is worth notified that this method add more 28.9% accuracy for SUPPORT and REFUTES but reduce 17% accuracy for NOT ENOUGH INFO. Overall, this ensemble model increase the accuracy from 38.1% to 41.7% for the test data.

| Actual | Prediction | | | Evaluation Metric | | |
|---|---|---|---|---|---|---|
| | SPT | RFT | NEI | PREC | REC | F1 |
| SPT | 407 | 241 | 1019 | 0.41 | 0.24 | 0.31 |
| | 683 | 282 | 702 | 0.42 | 0.41 | 0.41 |
| RFT | 252 | 372 | 1043 | 0.42 | 0.22 | 0.29 |
| | 420 | 578 | 669 | 0.47 | 0.35 | 0.4 |
| NEI | 339 | 273 | 1055 | 0.34 | 0.63 | 0.44 |
| | 536 | 364 | 767 | 0.36 | 0.46 | 0.4 |

Even though the ensemble model slightly increase the accuracy, but it still around the baseline. Thus we try another model with completely different features. We build an MLP model using embedding vectors of claim, embedding vectors of selected evidences, cosine similarity between claim and evidence as vectors, entities, and nouns for the features. This approach reach 44% label accuracy on development set. Even though this model return higher accuracy than the previous one, this model only applicable to predict SUPPORT and REFUTES as the F1 score for NOT ENOUGH INFO is only 0.1.

## 3. Experiments

Before we comes to our final approach, we have done massive experiments and encountered several challenges that lead us to more insights and further strategies. The following part discuss our experiments during the project.

### 3.1. Index Construction

Our initial trail was to build the index manually. However, the manual index takes about 200 to 400 seconds to return relevant documents per claim. Processing 14K test data will need more than two month to finish. Thus, we shifted to Information Retrieval engine using Xapian, which was the fastest among other libraries we tried (such as Whoosh).

Indexing can be done in document level or sentence level. Considering that one sentence is mostly not enough to describe the claim and it is highly correlated with other

sentences in the same document, we decided to do indexing based on document is the prefered one.

Xapian default similarity measure is BM25 and we keep on this setting because as we tried to change into TF-IDF, the performance on document selection is decreased by half compared to BM25. In order to improve the IR, we add more features to the documents. Xapian allows to store features for each document. We embed non-stopwords from the document sentences to be calculated in similarity measure. This setting turns to slightly increase the document selection performance.

Xapian allows add more features for similarity checking. For instance, we attempted to build an index based on named entity recognition. However, each wiki file took 15 minutes to index. As the dataset contain 109 files in total it would take more than 24 hours to complete. This would breach the time limit constraint of building a system that can be reconstructed, trained and evaluate the claims within 24 hours window.

### 3.2. Query Formulation

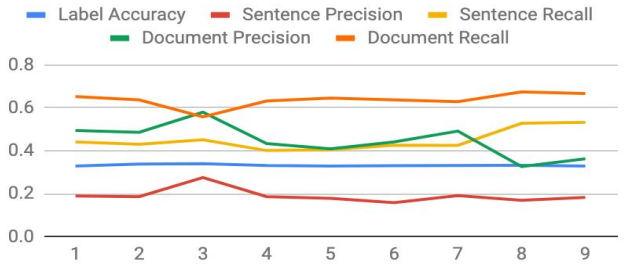
In order to improve the retrieval performance we tried to modify the query. Using raw claim for some cases cannot capture the real evidence documents. Claim "*Damon Albarn's debut album was released in 2011*", for instance, have evidence "*Damon_Albarn*", but in top 50 returns this document not exist. Removing punctuation only will turn "*Albarn's*" into "*Albarns*" which also lead to poor returns. Thus we normalize the claim by expand the contraction and remove the punctuations including the bracket symbols afterwards.

Subject of ownership can also be gained using entity extraction. We use spaCy large vocabulary Named Entity Recognition (NER) to extract the entities. However, using entity alone as the query not improve the retrieval documents. Whilst, query expansion by adding entities to normalized claim returns similar results with using the normalized claim only, but, require higher processing time. The better approach is using normalized claim with stopwords removal. For claim "*Savages was exclusively a German film*" without stopwords removal, the real evidence appear as the second rank, while with stopwords removal it appear as the top one.

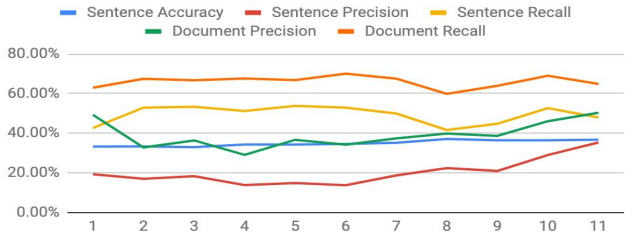
### 3.3. Document Selection and Sentence Selection

As re-ranking document retrieval return better result than IR, we tried several different setting for the title similarity threshold and selection priority. The first trial is used in the first submission, the second trial is add another layer after title similarity filtering. It is using AllenNLP textual entailment by taking top 5 documents that have entailment or contraction level above 0.7 based on the sense that these documents are either evidences for supports or refutes the claim. Unfortunately, this adjustment does not improve the results, instead, decrease the performance.

The third trial is by using similar setting as the first trial except taking the top 10 documents from IR instead of 2, then do sentence selection using a supervised machine learning model. This setting improve overall performance of the first approach except the document recall decreased by 10%. The next six trials using similar setting as the first approach by combining the similarity measure using entities and capitalized words [8,9], adjusting the similarity threshold to 0.3 [6] and limiting number of sentences at maximum 5 or 10 [4,5,7]. Overall, these adjustments do not improve the previous results. The issue is as recall increase, the precision goes down. It means that the majority of true evidences are included in the selected sentences but there are excessive irrelevant documents returned.




It is noticeable that claim and its correct evidence has high similarity in terms of tokens, entities, and capitalized words. Both document and sentence recall raise up to 67.45% and 53.32% respectively when filtering based on entities and capitalized words. Accordingly, in our final approach we combine these features for document selection and sort by claim and sentences similarity for sentence selection. The following chart shows the history of set the threshold for sentence selection before reach the final approach on the 11th trial. Similar to the previous approach, we encounter precision-recall trade-off. Louse filtering will end up in more sentences returned and decreasing precision.



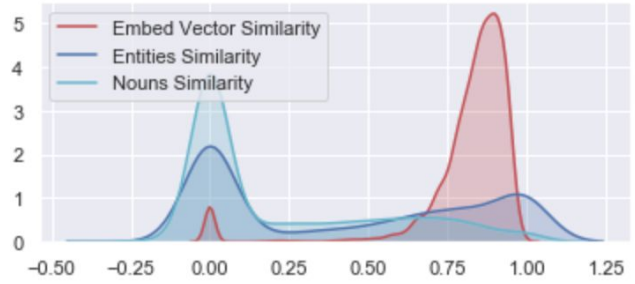

### 3.4. Claim Verification

The reason we use MLP Classifier is because it return better performance compared to Random Forest and Multinomial Naive Bayes. Whilst, choosing 1 hidden layer with 300 nodes is because as we tune this hyperparameters, we found that having 1 hidden layer with more nodes return better accuracy than having more layers with lower nodes, for example MLP model using 20 nodes returns the same accuracy with using 3 layers with 10 nodes per layer.

During the experiments we also tried to build BOW with and without stopwords. The results shows for every combination of different hyperparameter in MLP Classifier, the performance of BOW with stopwords outweigh BOW without stopwords. Furthermore, represent BOW in TF-IDF score also improve the results compared to word frequency. However, embed the TF-IDF to 2000 dimensions of dense matrix using Singular Value Decomposition (SVD) decrease the accuracy score by half. Similarly, since the class is

imbalance, we tried to train the model using oversampling and undersampling to get balance records for each class but we got decreasing performance. Thus, in our submissions we use TF-IDF matrix from BOW with stopwords as features from 140,000 random records of training set in actual proportion.

As mentioned in methodologies part, we also tried to build model with embedding vectors of claim, embedding vectors of selected evidences, cosine similarity between claim and evidence as vectors, entities, and nouns for the features. We found that the similarity features play an important role to the model, as when we exclude these features with only using the embedded vectors the performance depreciated. The distribution of similarities also shows two spikes which might be related to the class.




This model shows better performance compared the BOW words in terms of accuracy, however, this model is lack of NOT ENOUGH INFO predictions, as only 4% predictions on this class and the recall for this class only 0.1.

| | Precision | Recall | F1 |
|---|---|---|---|
| SUPPORTS | 0.45 | 0.5 | 0.47 |
| REFUTES | 0.44 | 0.76 | 0.56 |
| NOT ENOUGH INFO | 0.44 | 0.06 | 0.1 |

## 4. Error Analysis

The following is the error analysis for each of the three sub-modules of the system, that can be the basis for further improvements of the system.

### 4.1. Document Retrieval

During the training phase, we identified some documents that are not getting retrieved docs from IR fetch. These documents exist in our index but the IR engine cannot recognize them directly. Some of them are documents with apostrophe, such as:

"*Robert_J._O'Neill_-LRB-U.S._Navy_SEAL-RRB-*".

For the training phase, all claims that have evidence not found are excluded. Since, we want to avoid relevant document not returned because the claim has apostrophe, we normalize the query to expand any contraction.

### 4.2. Document and Sentence Selection

The re-rank document based on claim and document title similarity in general shows better performance than just rely on IR engine. However, there is a drawback when there are many documents have title similarities with the claim but they are not the actual evidence. For example, claim "*CBS is the network that aired The Millers*" has actual evidence "*The_Millers*" but our system pick "*CBS*" and "*List_of_The_Millers_episodes*" because these two documents title highly similar in terms of entities, nouns, and tokens matching with the claim. The actual evidence will appear in the document selection if we relaxed the threshold, however, it will return low precision.

### 4.3. Claim Verification

The ensemble model is only bag of words, thus, if a certain token from a claim is not found in the evidence, it tends to predict NOT ENOUGH INFO, while model with embedding vectors find relation between the similarity of words, if similar words is found it tends to predict SUPPORT. For example, for claim "*Saxony is in Ireland*" the actual label is REFUTES with evidence Saxony on sentence 0 "*The Free State of Saxony … is a landlocked federal state of Germany …*", the predicted evidence is Saxony on sentence 12 "*Old Saxony .. German … the Westphalian part of North Rhine-Westphalia*", the ensemble model predict as NOT ENOUGH INFO and the embed vector model predict as SUPPORT. It might be because German in evidence has similar vector with Ireland as they represent countries.

## 5. Future Work

Following our latest model, in future work we will train a binary classifier between SUPPORTS and REFUTES only and assign NOT ENOUGH INFO in sentence selection steps using another binary classifier to detect TRUE or FALSE evidence. We will add more features to the model to capture semantic distribution across claim and evidence, such as sequential POS tagging, relation extraction from claim and evidence, root words, synsets of keywords, and negation words. We might also use different Neural Network model.

## 6. Conclusion

In this project, we attempt to reach the baseline and ranked 132 for label accuracy and ranked 68 for sentence selection in Codalab. We do re-ranking for document selection by considering claim and document title similarities in terms of tokens, entities, and capitalized words. We found that combining BOW model and textual entailment model increase the accuracy. We also improve our claim verification model using more features such as claim and evidence embedding vectors, and their similarities as vectors, entities, and nouns which lead us to future work on training this model as binary classifier.

Through the process we learned that providing more evidence to a claim will lead to precision decline and eventually lead to misclassification. Choosing a classifier, tuning hyperparameters, text normalization, and word embedding also play important roles to the result performance and there is no fix steps for every case. In most cases, stopwords removal and SVD might helps but not in our trials. Handling imbalance class also not necessarily be done if with the actual proportion lead to better result. In conclusion, there are many insights can be derived from this project and such future work is needed since this fact verification is beneficial for many aspects in human life.

**References**

Hanselowski, A., Zhang, H., Li, Z., Sorokin, D., Schiller, B., Schulz, C., & Gurevych, I. (2018). UKP-Athene: Multi-Sentence Textual Entailment for Claim Verification. *arXiv preprint arXiv:1809.01479*.

Thorne, J., Vlachos, A., Cocarascu, O., Christodoulopoulos, C., & Mittal, A. (2018). The Fact Extraction and VERification (FEVER) Shared Task. *arXiv preprint arXiv:1811.10971*.

Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.