The Copenhagen Team Participation in the Factuality Task of the Competition of Automatic Identification and Verification of Claims in Political Debates of the CLEF-2018 Fact Checking Lab

Dongsheng Wang[*], Jakob Grue Simonsen[*], Birger Larsen[†], Christina Lioma[*]

(*) Department of Computer Science University of Copenhagen (DIKU))
{wang,simonsen,c.lioma}@di.ku.dk
(†) Department of Communication, Aalborg University Copenhagen
birger@hum.aau.dk

Abstract. Given a set of political debate claims that have been already identified as worth checking, we consider the task of automatically checking the factuality of these claims. In particular, given a sentence that is worth checking, the goal is for the system to determine whether the claim is likely to be true, false, half-true or that it is unsure of its factuality. We implement a variety of models, including Bayes, SVM, RNN, to either step-wise assist our model or work as potential baselines. Then, we develop additional multi-scale Convolutional Neural Networks (CNNs) with different kernel sizes that learn from external sources whether a claim is true, false, half-true or unsure as follows: we treat claims as search engine queries and step-wise retrieve the top-N documents from Google with as much original claim as possible. We strategically select most relevant but sufficient documents with respect to the claims, and extract features, such as title, total number of results returned, and snippet to train the prediction model. We submitted results of SVM and CNNs, and the overall performance of our techniques is successful, achieving the overall best performing run (with lowest error rate 0.7050 from our SVM and highest accuracy 46.76% from our CNNs) in the competition.

Keywords: political debates, RNN, CNN, fact checking

1 Introduction

The Copenhagen team participated in both Tasks 1 and 2 of the CLEF-2008 Fact Checking Lab for the English language. This report details our methods and results for *Task 2*, the task description paper by the organizers with all background and details specified can be found in [2]. Our participation in Task 1 is described in [3].

Given a set of political debate claims that have been already identified as worth checking, the aim of the second task is to check the factuality of these claims. In particular, given a sentence that is worth checking, the goal is for the system to determine whether the claim is likely to be true, half-true, false or that it is unsure of its factuality.

One of the two examples given by organizers [8] is shown in Table 1, where Hillary Clinton mentions Bill Clinton's work in the 1990s, followed by a claim made by Donald Trump stating that for president Clinton approved the North American Free Trade Agreement (NAFTA). This last statement by Trump is judged to be HALF-TRUE because it was George H.W. Bush who signed the approval for NAFTA, but Bill Clinton who signed it into law.

Table 1. Example of a spurious claim

Speaker	Sentence
CLINTON:	I think my husband did a pretty good job in the 1990s.
CLINTON:	I think a lot about what worked and how we can make it work again...
TRUMP:	Well, he approved NAFTA... (HALF-TRUE)

As CLEF provides limited data (only 82 unique claims with labels), but the task of fact checking relies on labeled data to train prediction models, finding suitable datasets for training is the first basic step. Furthermore, the task at hand is more complex than traditional binary prediction (True/False) as graded truth values must be predicted, including the difficult "Half-True". There are primarily three objectives that we take into consideration:

1. Select external claims with labels and suitable proportion of samples
2. Retrieving most relevant but suitable amount of external sources (documents) for claims
3. Find the best models and parameters and tune them to their best performance

The three objectives are met by proceeding in a stepwise manner. Selecting external claims of high quality is the basis of the following steps. The multiple labels and their proportional samples have to be taken into account when selecting datasets with different labeling. Subsequently, retrieving most relevant but adequate documents for these claims are significant to support the building of training models. Finally, selected features of documents should be fitted well into different models, of which the parameters should be tuned to improve the final results.

2 Approaches Used and Progress Beyond State-of-the-Art

Our approach is as follows: we use a step-wise modeling from data selection, preprocessing, retrieval, to training modeling, with the aim of choosing a suitable proportion of samples with labels and supporting documents that we are going to employ. Specifically, we take advantage of a simple Bayesian model, to analyze label impact and data sufficiency in the data processing stage, and stepwise search in the stage of supporting document retrieving.

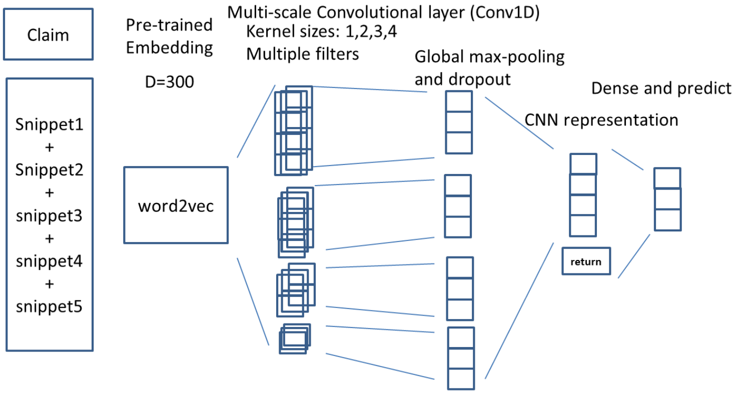
For training the model, we employ CNN and RNN models, as well as an SVM. RNN model is employed in similar tasks such as the work of detecting rumors from microblogs by Ma et al. [6] through capturing the variation of contextual information of relevant posts over time in microblogs. Closer to our aims, Karadzhov et al. [4] investigate a fact checking task, and we implement a similar framework as shown in Figure 2; we use two simplifications compared to [4]: (a) we only use 5 Google snippets while the original author uses 4 units consisting of one Google snippet, one Bing snippet and two triplets of rolling sentences from Google and Bing respectively; and (b) we only calculate one similarity, namely pairwise TF-IDF cosine similarity, whereas [4] calculates the average of cosine with TF-IDF, cosine over embeddings, and containment.

CNNs are adopted in the sentence-level classification by Kim et al. [5] and they have been demonstrated to improve performance on NLP classification tasks. In our CNN model, as shown in Figure 1, inspired by [10], we employ multi-scale CNNs with different kernel sizes to overcome the drawback of simple convolutional kernel with fixed window size over encoded semantics of documents. It is hard to determine window size using simple convolutional kernels, because small window normally requires deeper networks to gain critical information and large window sizes result in loss of local information. Therefore, multi-scale CNNs, together with the other feature (total return), are designed to represent the comprehensive contextual information of the text. Specifically, we encode the semantics with word2vec [7] for documents of concatenated snippets on the first layer into low-dimensional vector. Then, we perform multi-scale CNNs with different kernel sizes on the second layer over the embedded word vectors. In our experiment settings, we concatenate four CNNs with 1, 2, 3 and 4 kernel sizes respectively, followed by max-pooling layer and dropout after each of them. We set the channel as static word vectors. In CNN representation layer, we add total return of the first search (step-wise search is discussed in Section 3.2) for each claim as additional information. Because the total return has a large numeric range, we discretize it into eight equally-sized categories based on the statistics of training samples.

While each step of our approach uses only simple, well-known methods, our progress beyond state-of-the-art methods consists of the *combination* of the following:

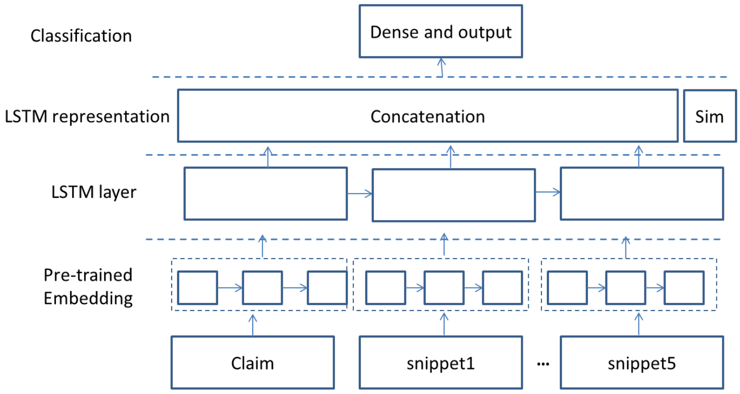
1. We use step-wise modeling, instead of using a mixed model in the final step, i.e., we use traditional Bayes models for data prepossessing, including data selection (label mapping) and external source analysis (sufficiency analysis), and then build a CNN model based on previous conclusion.
2. We employ step-wise searching in retrieving supporting documents with as much of the original claim as possible while strategically retrieving enough documents, instead of just using keywords.

Fig. 1. Our CNN model framework



3. we employ multi-scale CNNs with multiple kernel sizes, together with discrete total return, to represent the contextual information. We assume that multi-scale CNNs can obtain comprehensive information and the total return of a claim represents the intensity of attention, which to some degree reflect its hidden status.

Fig. 2. RNN model framework



3 Resources Employed

We describe how to collect claims with labels from Politifact in Section 3.1, and how we retrieve supporting documents for these claims in Section 3.2, and we give a short description concerning the word embedding we utilize in Section 3.3.

3.1 Claims with labels from Politifact

Due to the small dataset of claim samples (a total of 82 unique claims with labels) provided by CLEF-2018 Fact Checking Lab, we use Politifact as an external source to collect claims and their labels. Specifically, we crawl Politifact Truth-O-Meter statements from www.politifact.com that are operated by editors and reporters from the Tampa Bay Times. For the Truth-O-Meter webpage, Politifact staffers research U.S. politics statements and label them as "true", "mostly true", "half true", "mostly false", "false", "and pants on fire" (the latter for outrageously false claims). We obtain a total of 4,604 statements/claims from Politifact as demonstrated in Table 2.

Table 2. Distribution of labels of claims

True	Mostly True	Half True	Mostly False	False	Pants on Fire	All
654 (14.2%)	863 (18.7%)	974 (21.2%)	776 (16.9%)	911 (19.8%)	426 (9.3%)	4,604

The task of CLEF-2018 Fact Checking Lab requires us to predict claims as one of the three labels: "true", "false" and "half-true". Therefore, we map the six Politifact labels into three categories and remove some ambiguous labels. Table 3 shows three examples of label mapping; for Map1, we map all six into three labels; remove Mostly False for Map2; and Mostly-true as well in Map3. In experiment part (Section 4.1), we would compare the performance for each mapping and obtain the best one as training dataset.

Table 3. Different Combination of Label Mapping

Mapping	FALSE	TRUE	HALF-TRUE
Map1	False,Pants on Fire!,Mostly False	True,Mostly True	Half-True
Map2	False,Pants on Fire!	True,Mostly True	Half-True
Map3	False,Pants on Fire!	True	Half-True

In addition, we try to discover how many overlaps does Politifact have with that in test set. If we check the exactly same claims, no claims are found, i.e., no overlap exists. If we use Levenshtein distance to detect similar claims, there are still no same claims exist when similarity is set below 0.8, whereas there are only three claims that are, to some degree, similar when the similarity is set below 0.8.

It is noted that of all the retrieved URLs, there were a total of 1,310 bad URLs out of 84,451 URLs (A ratio of 1.55%) in our training dataset. For model building and training with our existing dataset, we use the politifact dataset without further processing or filtering. For testing data from CLEF, we used the url-filters function provided by CLEF to filter bad urls when retrieving supporting documents from Google, and output the prediction result. According to our internal testing, the performance does not seem to be negatively impacted with or without bad URLs in training dataset, and is sometime even slightly better after filtering.

3.2 Documents Retrieval from Search Engine

We retrieved supporting documents and texts in order to train the prediction models, in addition to the claim texts themselves. To that end, we retrieve top-N documents from Google. Compared to the classical approach of analyzing and shortening claims into keywords, we used a step-wise searching method to maintain the semantics as much as possible.

The reason for this is that we conjecture that using the whole sentence could keep more of the original semantics, including speaking habits and commonly-used sequences of words. We thus use each whole claim verbatim as a Google query, at the risk of retrieving only few documents. We subsequently apply step-wise searching to fill up a list of documents as follows. First, we initialize a set with zero documents for each claim. We then use the whole claim as a query to retrieve documents and populate the set with the results. If the set has less than N documents (N=20 in our concrete experiments), we remove the stop words of the claim and search again, populating the set with retrieved documents. If the set contains fewer than N documents, we search again using only the nouns, verbs and adjectives obtained by using part-of-speech (POS) (obtained with the Stanford POS tagger [9]), populate the set in the same way as with the second search. We ended up being able to retrieve 20 documents for each of the claim. It is noted that we do not necessarily use all of them as we re-rank the documents with cosine similarity and employ, for example, top 4 or 5 snippets among them.

3.3 Pre-trained embedding

For CNN and RNN, We employ existing pre-trained word vectors - word2vec [7] for our word embedding. word2vec is published by Google who trained it on 100 billion words of Google news with continuous bag-of-words model and generated the vectors of 300-dimension.

4 Analysis of the Results

In Section 4.1 and 4.2, we perform some experiments with a Bayesian classifier to investigate how to map the six labels into three, and determine how many documents (snippets) we need to fit models. We do not use more advanced classifiers such as RNNs or CNNs to conduct this analysis as the word embedding layer is hidden, hard to explain, and the proper neural network layer are sensitive to parameters rather than semantics of texts. Conversely, it is usually easier to understand how a trained Bayesian classifier based on bag of words and n-gram reflect the semantics of texts in a simple and straightforward way. In Section 4.3, we give the comparison of performance of different models on the test dataset.

4.1 Label mapping

The six Politifact labels must be mapped into three: True, False, and Half-True. Some labels are ambiguous, e.g., Mostly-True can be either True or Half-True. Therefore, we tune the mapping using a Bayes classification model on three combination cases listed in Table 3. As shown in Figure 3, Map3 has the same highest accuracy with Map1, the highest macro F1 and the highest macro recall. Therefore, we apply Map3 mapping case as our training data. We discard the claims with labels not listed in Map3, and the new statistics is shown in table 4, which is applied to all other models as well.

Fig. 3. Performance with different Label Mapping Cases

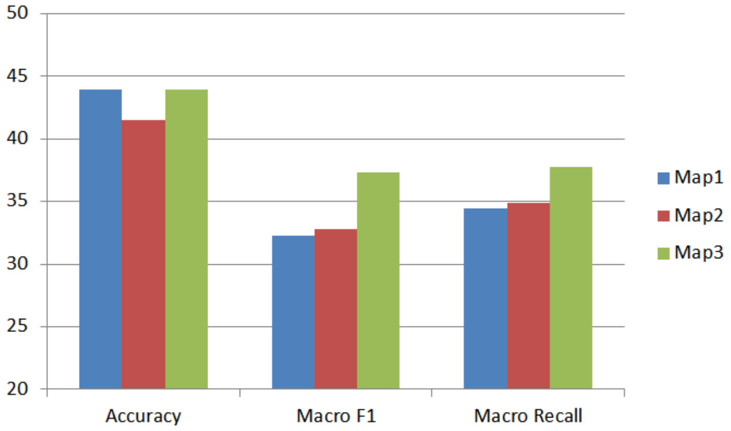


Table 4. Statistics of Selected Samples

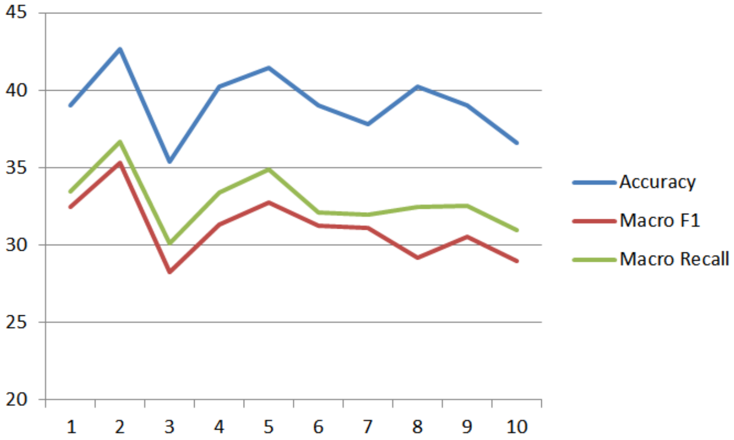
	FALSE	TRUE	HALF-TRUE	Total
Map3	**False,Pants on Fire!**	**TRUE**	**Half-True**	\
Number	**1,305**	**1,473**	**941**	**3,719**

4.2 Semantic Contents Sufficiency Analysis

We manipulate the number of document snippets that are concatenated and compare their performance (accuracy, macro F1, etc.) to determine the number of snippet texts needed. We rank the documents according to their pairwise TF-IDF similarity with the claims and select the top-N (we test N=1 to N=10 in our experiments) to concatenate.

As shown in Figure 4, using two snippets leads to the highest performance, 5 snippets the second-best, and 4 and 8 the third-best. In short, the rank of ”2,5,4,8, etc.” is the order of numbers of snippets we learned that we can refer to utilize. As we know, training deep learning model is time-consuming, such analysis narrows the scope of choices and enables us to focus on parameters of models themselves. However, note that the results by number of snippets are quite unstable, and that no firm conclusions can be drawn. In our experiment, we primarily conduct our experiment on 2 and 5 snippets (highest and second highest) and attempt to obtain the models of parameters and number of snippets with the best performance.

Fig. 4. Performance with different snippet number



4.3 Prediction comparison

For Bayes and SVM classifiers, we employ the Bag of Words (BOW) model for English texts, tokenizing them and removing stop words; also, we adopt TF-IDF (Term Frequency times inverse document frequency) for term weight. We use grid search to tune the parameters of each model. We rank the documents of each claim according to their similarity with the claim and concatenate the first five snippets as a whole document. As shown in Table 5, Naive bayes with grid search could reach its best performance to 53.98% and 43.90% on Politifact samples

(20% of Politifact samples=743 samples) and CLEF (82 samples), respectively. We produce two best models, with CNNs and RNN, on Politifact and CLEF test respectively, titled as CNN1, CNN2, RNN1, RNN2, shown in the table. For CNN and RNN, we observe that RNN has worse performance than CNN on Politifact samples, with 47.22% and 46.88%, respectively. The performance on CLEF samples with 45.12% and 46.34% is similar. In contrast, using CNN results in an accuracy of 55.56% and 51.56%, respectively, on Politifact samples, but only 42.68% and 48.78% on CLEF samples.

We submitted the SVM+Gridsearch and CNN models because SVM is more stable across most test cases, and CNN has a relatively good overall performance, but is less stable. We run the two models on the final test dataset without label from CLEF and output the prediction result. The test result from CLEF is shown in Table 6. Within our two groups of results, we observe that for metric MAE (mean absolute error) SVM outperforms CNN while for accuracy CNN2 outperforms SVM. However, either one group of our results outperforms those of all the other teams for each single metric.

Table 5. Test Result with different models (%)

Test dataset	**Politifact**	**CLEF (82 samples)**		
Metrics	Accuracy	Accuracy	Macro F1	Macro Recall
zeroR (majority voting)	50.00	39.60	\	\
ngram+SVM (baseline)	50.64	39.02	29.97	31.87
ngram+SVM+GridSearch	53.85	43.90	30.14	33.02
Naive Bayes+GridSearch	53.98	43.90	36.10	37.47
RNN+LSTM (best on Politifact)	**47.22**	45.12	35.95	36.66
RNN+LSTM (best on CLEF)	46.88	**46.34**	37.75	39.56
CNN1 (best on Politifact)	**55.56**	42.68	28.42	32.67
CNN2 (best on CLEF)	51.56	**48.78**	39.66	39.80

Table 6. Evaluated Result by CLEF

Copenhagen	MAE	Macro MAE	ACC	Macro F1	Macro Recall
primary(SVM)	0.7050	0.6746	0.4317	0.4008	0.4502
cont.(CNN2)	0.7698	0.7339	0.4676	0.4681	0.4721

There are several further empirical phenomena evident from our experiments:

1. The RNN model is more unstable than other models and sensitive not only to parameters but also to epochs.
2. The traditional models, Bayes and SVM, sometimes have worse performance than the neural network-based approaches, but are much more robust in terms of performance.

3. We originally conjectured that claims in documents could be concerned with temporal information which could be exploited well by an RNN model. Hence, we also try to re-rank the documents of a claim according to year, and fit on RNN model. This does not improve performance. One possible reason is that some documents do not have time information, and placing them in a ranked list (in front or rear) just introduces uncertainty. Another reason might be that we adopt only a few documents so the ranking is not as apparent as we assumed.

5 Perspectives for Future Work

Our current work only employs lexical and syntactic context. In future work, we plan to add information about semantic structures and argumentation flow; we believe that this will aid our methods in identifying some of the most egregious common examples of poor reasoning or argumentation (e.g., logical fallacies). One similar work that could be referred to is by Ba el at. [1] who extract entities and relations from web and Twitter and gathers the conflicting information. Secondly, while we have found that using combinations of *simplifications* of several methods found in the literature, we aim to investigate whether using tuned versions of the original methods (e.g., [4]) may improve our results.

6 Acknowledgement

This work is funded by the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 721321.

References

1. M. L. Ba, L. Berti-Équille, K. Shah, and H. M. Hammady. VERA: A platform for veracity estimation over web data. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11-15, 2016, Companion Volume*, pages 159–162, 2016.
2. A. Barrón-Cedeño, T. Elsayed, R. Suwaileh, L. Màrquez, P. Atanasova, W. Zaghouani, S. Kyuchukov, G. Da San Martino, and P. Nakov. Overview of the clef-2018 checkthat! lab on automatic identification and verification of political claims, task 2: Factuality. In L. Cappellato, N. Ferro, J.-Y. Nie, and L. Soulier, editors, *CLEF 2018 Working Notes. Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum*, CEUR Workshop Proceedings, Avignon, France, September 2018. CEUR-WS.org.
3. C. Hansen, C. Hansen, J. G. Simonsen, and C. Lioma. The Copenhagen Team Participation in the Check-Worthiness Task of the Competition of Automatic Identification and Verification of Claims in Political Debates of the CLEF-2018 Fact Checking Lab. Technical report, CLEF Fact Checking Lab, 2018.
4. G. Karadzhov, P. Nakov, L. Màrquez, A. Barrón-Cedeño, and I. Koychev. Fully automated fact checking using external sources. In R. Mitkov and G. Angelova,

editors, *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017, Varna, Bulgaria, September 2 - 8, 2017*, pages 344–353. INCOMA Ltd., 2017.

5. Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751, 2014.
6. J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha. Detecting rumors from microblogs with recurrent neural networks. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 3818–3824. AAAI Press, 2016.
7. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
8. P. Nakov, A. Barrón-Cedeño, T. Elsayed, R. Suwaileh, L. Màrquez, W. Zaghouani, P. Gencheva, S. Kyuchukov, and G. Da San Martino. Overview of the CLEF-2018 lab on automatic identification and verification of claims in political debates. In *Working Notes of CLEF 2018 – Conference and Labs of the Evaluation Forum*, CLEF '18, Avignon, France, September 2018.
9. K. Toutanova and C. D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13*, EMNLP '00, pages 63–70, Stroudsburg, PA, USA, 2000. Association for Computational Linguistics.
10. S. Wang, M. Huang, and Z. Deng. Densely connected cnn with multi-scale feature attention for text classification. 2018.