TheEarthIsFlat's Submission to CLEF'19 CheckThat! Challenge

Luca Favano[1], Mark J. Carman[2], and Pier Luca Lanzi[3]

Politecnico di Milano MI 20133, Italy
[1] luca.favano@gmail.com
[2] mark.carman@polimi.it
[3] pierluca.lanzi@polimi.it

Abstract. This report details our investigations in applying state-of-the-art pre-trained Deep Learning models to the problems of Automated Claim Detection and Fact Checking, as part of the CLEF'19 Lab: *CheckThat!: Automatic Identification and Verification of Claims*. The report provides an overview of the experiments performed on these tasks, which continue to be extremely challenging for current technology. The research focuses mainly on the use of pre-trained deep neural text embeddings that through transfer learning can allow for improved classification performance on small and unbalanced text datasets. We also investigate the effectiveness of external data sources for improving prediction accuracy on the claim detection and fact checking tasks. Our team submitted runs for every task/subtask of the challenge. The results appeared satisfactory for task 1 and promising but less satisfactory for task 2. A detailed explanation of the steps performed to obtain the submitted results is provided, including comparison tables between our submissions and other techniques investigated.

Keywords: Automated Fact Checking · Claim Detection · Text Classification · Natural Language Processing · Deep Learning

1 Introduction

In this report we describe our efforts to use state-of-the-art pre-trained deep neural text embeddings for tackling the different subtasks of the *CheckThat! challenge* [6]. In order to achieve good results, a great number of experiments were performed. In the following sections we provide descriptions and results for the most interesting of these experiments in the hope of inspiring future research in this area. In Section 2 we will explain all the steps that brought to our final submission for Task 1, from the choice of the architecture to the fine-tuning of the chosen setup. In Section 3 we explain the text pair classification approach that we applied for the subtasks of Task 2.

Table 1: Example sequence of utterances and their corresponding binary labels (*check-worthy* or *not check-worthy*) from the *CheckThat! challenge* Task 1 dataset

Speaker	Sentence	Label
Sanders	And what has happened there is absolutely unacceptable.	0
Maddow	Senator, thank you.	0
Todd	Secretary Clinton, let me turn to the issue of trade.	0
Todd	In the '90s you supported NAFTA.	1
Todd	But you opposed it when you ran for the president in 2008.	1

2 Task 1 - Check-Worthiness

The first task [1] for the CheckThat challenge involved classifying individual statements within political debates as *check-worthy* (i.e. constituting a claim that is worth fact checking) or *not check-worthy*. The training data consisted of 19 debates, while the test data contained seven. An example section from one of the debates[1] is shown in Table 1. Note that each debate is a dialog with the speaker information available for each utterance.

2.1 Preliminary Experiments

Recent years have seen a proliferation of pretrained-embeddings for language modeling and text classification tasks, starting from basic word embeddings such as word2vec [12] and GloVe [14], and moving to sub-word and character-level embeddings like FastText [11]. More recently pre-trained deep networks have become available, which make use of BiLSTM [8] or self-attention layers [16] to build deep text processing models like ELMo [15] and BERT [5]. These models offer improved transfer learning ability, taking advantage of massive corpora of unlabeled text data from the Web to learn the structure of language, and then leveraging that knowledge to identify better features and improve prediction performance on subsequent supervised learning tasks.

In this work, we make use of a number of state-of-the-art pre-trained models for text-processing, namely: BERT [5], ELMo [15], Infersent [4], FastText [11], and the Universal Sentence Encoder (USE) [3].

When competing in the challenge we first ran a preliminary experiment over validation data comparing the performance of these toolkits in order to decide which one to use for our submission. We repeated this comparison after the annotated test set for the challenge was published, so that we could provide results on the held-out test data. Those test results for Task 1 are reported in Table 2. Note that default (hyper)parameters were used for each system, with the exception of the number of training steps (or epochs), which was set based on validation performance.

[1] Sample sentences extracted from the file "20160209-msnbc-dem".

Table 2: Performance comparison on Task 1 data only

Toolkit	MAP	RR	R-P
BERT [5]	0.0824	0.3112	0.0776
ELMo [15]	0.0587	0.3466	0.0729
Infersent [4]	0.1057	0.2503	0.1034
FastText [11]	0.1445	**0.5303**	0.1545
USE [3]	**0.1871**	0.3679	**0.2071**

Table 3: (Hyper)Parameter settings used for the Universal Sentence Encoder

Parameter	Value
Total Steps	600
Optimizer	Adam
Width of Hidden Layers	100/8
Activation Function	ReLU
Dropout	0.1
Learning Rate	0.005
Trainable Parameter	False

2.2 Modifying the Training Data

Results of the preliminary experiment indicated that the Universal Sentence Encoder (USE) was a model that could provide reasonable performance for the claim detection task. We then investigated a number of different settings for how to train a USE-based classifier and how to modify the training dataset in order to improve prediction performance. The modifications to the training dataset considered included appending speaker information or previous utterances to the input and also the use of external training data.

For the classification task, the network architecture used was to append a fully connected Feed-Forward (FF) Neural Network with two hidden layers to the output from the Universal Sentence Encoder. The training (hyper)parameters for the network were set to the values shown in Table 3. Note that the weights of the USE encoding were not fine-tuned[2] during training of the classifier due to the small quantities of labelled training data available.

The following experimental setups were evaluated. We report results for each setting on the test data (not available at the time of run submission) in Table 4.

1. Training on Task 1 dataset only, using each individual sentence only as the input text.
2. Same as setup 1, but concatenating the speaker information to the sentence text.

[2] Investigations with the parameter *Trainable* set to *true* resulted in degraded performance.

3. Same as setup 1, but using as input the concatenation of the two previous sentences with the current sentence.
4. Same as setup 1, but applying basic text pre-processing, in which contractions in the text are expanded and the text is stripped of accented characters, special characters or extra white spaces, and then converted to lower-case.
5. Same as setup 1, but activating the *Trainable* parameter of the USE-module to fine-tune the weights of the sentence encoder.
6. Supplementing the Task 1 dataset with *additional positive examples* extracted from the LIAR dataset [17]. The LIAR dataset contains a set of political sentences from various sources that have been fact-checked by PolitiFact[3] and assigned a truth label. It is safe to assume that all the sentences included in the LIAR dataset were once considered worthy of fact checking. Based on this assumption all the sentences in the dataset make for a valid set of additional positive instances for the fact checking task. Moreover there is a strong motivation for adding positive examples to the Task 1 training set, since the training data is highly skewed toward the negative class with only a small percentage of positive training instances. An obvious limitation of this idea is that by adding only positive instances which come from a different source from the training data (and therefore may not share the same vocabulary distribution), we may simply end up training the classifier to distinguish between instances from the two datasets (the Task 1 political debate instances and the LIAR fact-checked claims dataset).
7. Training first on the LIAR dataset [17], but keeping the 0 and 1 labels the same as they were in the original LIAR dataset (where 1 indicates a false statement and 0 indicates a true statement), and then train again on Task 1 dataset.
8. Training on a much larger Headlines+Wikipdia dataset consisting of one million headlines from news articles sourced from an Australian news source[4] and one million randomly chosen sentences from the content of Wikipedia articles[5]. The assumption here is that random chosen sentences from Wikipedia are generally not making claims nor worth fact-checking, while headlines from news articles are more likely to state a claim and are interesting and therefore likely worth fact checking. After first training on the 2 million sentence corpus, we then further train (fine-tune) the model on the Task 1 dataset.

We note from Table 4 that none of the tested modifications to the training data resulted in improvements over the basic USE-based classifier. Of all the techniques, the most interesting appears to be that of adding millions of positive and negative examples from the Headlines+Wikipedia dataset, which caused relatively small degradation in Average Precision (MAP) while providing a marked increase in Reciprocal Rank (RR). We leave to future work an investigation of

[3] https://www.politifact.com
[4] https://www.kaggle.com/therohk/million-headlines
[5] https://www.kaggle.com/mikeortman/wikipedia-sentences

Table 4: Performance comparison between different setups

Setup	Modification to Training Data	MAP	RR	R-P
1	-	**0.1821**	0.4187	0.1937
2	Include speaker name	0.1687	0.3206	**0.2054**
3	Include 2 previous sentences	0.1255	0.3123	0.1735
4	Pre-process text (case-folding, etc.)	0.1676	0.3466	0.1976
5	Fine-tune encoder weights (*Trainable=true*)	0.1294	0.3729	0.1495
6	Add external data from LIAR [17] as positive	0.1262	0.1698	0.1487
7	Add external data from LIAR [17] as true/false	0.1288	0.3884	0.1333
8	Add external data from Headlines+Wikipedia	0.1694	**0.4441**	0.1793

Table 5: Performance comparison between the two different Universal Sentence Encoder (USE) models available

Model	MAP	RR	R-P
Standard: Deep Averaging Network	0.1597	0.1953	**0.2052**
Large: Transformer Network	**0.1821**	**0.4187**	0.1937

why that was the case and whether modifications to that dataset and its use could result in positive gains in MAP.

2.3 Comparing Different Encoder & Discriminator Architectures

The Universal Sentence Encoder (USE) offers two different pre-trained models that differ in their internal architecture. The standard USE module is trained with a Deep Averaging Network (DAN) [10], while the larger version of the module is trained with a Transformer [16] encoder.[6]

Performance for the two versions of the USE encoder on the test data are shown in Table 5. We note a much higher MAP value for the larger, transformer-based model.

In order to provide a discriminative model able to predict check-worthiness labels, two different network architectures have been layered on top of the USE architecture. The relative performance of the two models is shown in Table 6, and their descriptions are as follows:

1. The architecture used to produce most of the results in this report is a Feed Forward Deep Neural Network (FF-DNN) with two hidden layers, obtained by using the TensorFlow DNNClassifier component.
2. A second architecture consists of a dense layer with a ReLU [13] activation function, followed by a softmax layer allows to categorize the results. This

[6] A third version of the encoder, called "lite", is specifically designed for systems with limited computational resources, and thus was not investigated here.

architecture was implemented in Keras[7] applying a lambda layer to wrap the USE output.

Performance for the TensorFlow implementation (on the validation data) outperformed the Keras ReLU architecture, so we continued with that model in the other experiments.

Table 6: Performance comparisons using different architectures

Architecture	MAP	RR	R-P
FF-DNN	**0.1821**	**0.4187**	0.1937
ReLU [13]	0.1703	0.2238	**0.1988**

In order to decide how many steps to train each model for, we examined performance of the models against the number of training steps on individual debates from the training data as shown for the Large USE model in Table 7. For that particular model we decided to train the model for only 600 steps based on the average results across the training debates.

Table 7: Average precision scores on individual debates from the validation data, computed while training the USE Large FF-DNN model for a given number of steps

Steps	Trump-Pelosi	Trump-World	Oval-Office	Average
600	**0.60**	0.36	0.22	**0.393**
1200	0.53	**0.41**	0.22	0.387
1500	0.57	0.26	**0.26**	0.363
3000	0.48	0.28	0.20	0.320

2.4 Submitted Runs for Task 1

For the submitted runs we made use of both the *standard* and *large* USE architectures compared in Table 5. The standard USE model has been used for the first two runs: *Primary* and *Contrastive 1*, while the large USE model was used for *Contrastive 2*. Table 8 contains the results for the submitted runs[8]. The difference between the first two runs, which both use the standard USE model, is that for the first we used the Adagrad optimiser and a feed-forward network with two hidden layers of size 512/128 while for the second we employed the

[7] https://keras.io
[8] Note that some values are the same as Table 5.

Adam optimiser with two hidden layers of size 100 and 8. We note that our last run (Contrastive 2) obtained the best MAP score over all runs submitted by any team for Task 1.

Table 8: Scores for TheEarthIsFlat's official submissions to the challenge.

Submission	Model	Parameters	MAP	RR	R-P
Primary	Standard FF-DNN(512/128)	Adagrad 1500 steps	0.1597	0.1953	**0.2052**
Contrastive1	Standard FF-DNN(100/8)	Adam 1500 steps	0.1453	0.3158	0.1101
Contrastive2	Large FF-DNN(100/8)	Adam 600 steps	**0.1821**	**0.4187**	0.1937

The USE standard model had been chosen as the primary run because it had provided better peak results during training, while the large model provided more stable results. Note the results on the training data shown in Table 9, where the standard model outperformed the large model on two of the three debates used for training.

Independently from the model used, we see that there is large variation in the performance across the debates in the training set. Dealing with such large variation effectively is something that ought be addressed in future work. We note that on the test data, where the average MAP value is around 0.18, the average precision across the individual debates varies from 0.05 (for the 2015-12-19 debate) to 0.5 (for the 2018-01-31 debate).

3 Task 2 - Evidence and Factuality

The second task of the challenge [9] contains multiple subtasks which together form a path that aims at automating the fact-checking process. Given a claim and a set of the web pages, the subtasks consist of:

1. Ranking the web-pages based on how useful they are to assess the veracity of the claim.
2. Labelling the web-pages based on their usefulness into four categories: *very useful, useful, not useful, not relevant.*
3. Labelling individual passages within those pages that are *useful* for determining the veracity of the claim.
4. Labelling the claims as *true* or *false* given the discovered information.

Unlike Task 1 for which all the data was written in English, for Task 2 all content was written in Arabic. We generally worked directly with the Arabic text but also experimented with translating the content into English as discussed below.

Every subtask has been tackled using a similar setup: after processing the data to obtain a dataset that consists of two strings of text and a label to

Table 9: Scores evaluated on a subset of debates from the validation set

Model	Trump-Pelosi	Trump-World	Oval-Office
Standard: Deep Averaging Network	**0.580**	**0.561**	0.294
Large: Transformer Network	0.511	0.495	**0.376**

predict, we feed this pairs into a pre-trained BERT model [5] that we train to classify the relationships between the two texts. In some cases, we have also investigated adding external data that could be useful, given that the datasets for the subtasks were extremely small.

3.1 Task 2A and 2B – Determining Relevant Web-pages

For the first two subtasks we used an almost identical approach: We extracted the claim text and associated with each web page text using the Beautiful Soup parser[9] to remove HTML markup. The training sets then consisted of 395 labelled text pairs (claims, corresponding webpages and relationship labels).

A set of experiments on the dataset were performed using a small portion of the training data as a validation set. The accuracy results in Table 10 have been averaged over three runs to account for the variation due to very small training/validation sets. The techniques investigated were the following:

1. BERT model is trained on the Task 2-AB dataset.
2. BERT model is trained on external data using a dataset that was previously used for stance detection for the FakeNewsChallenge [7] challenge.
3. BERT model has been first trained on the FakeNewsChallenge dataset then on the Task 2-AB dataset.
4. The Task 2-AB dataset has been translated to English before feeding it to the model as in 1.

Given that training BERT over large sections of text has very large memory requirements, the standard pre-trained BERT model was used instead of the biggest one available[10]. This limited the text sections to be no more than 100 to 150 words. BERT automatically reduces the information in longer context windows such that the this limit is enforced, implying that some information is necessarily lost from the text of longer webpages.

We observe in Table 10 improved performance using the FakeNewsChallenge dataset and translating the Arabic text to English, but caution that the results are subject to significant variation due to small sample sizes.

The ranking for subtask 2A was computed using the predicted confidence value with which the pages were being classified as useful. Analyzing the Challenge's "Results Summary", it can be noted that while the system learnt to

[9] https://www.crummy.com/software/BeautifulSoup/

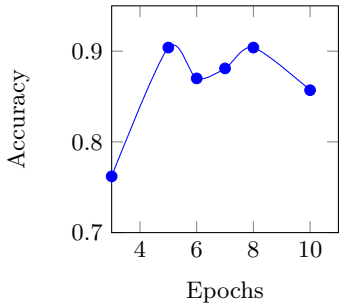
[10] We conjecture that the use of the bigger BERT model would have increased performance on these subtasks.

Table 10: Accuracy on the validation data of predicting the usefulness of a webpage for subtasks A and B. (The same prediction model was used for both tasks.)

Setup	Treatment	Average Accuracy
1	Use Task 2-AB dataset	0.481
2	Use FakeNewsChallenge data	0.502
3	Use FakeNewsChallenge + Task 2-AB data	**0.575**
4	Translate Task 2-AB data to English	0.527

Table 11: Accuracy results on validation data for subtask C while varying the number of training epochs

Epochs	Accuracy
3	0.762
5	**0.904**
6	0.870
7	0.881
8	**0.904**
10	0.857



classify *not relevant* and *not useful* pairs of texts, it was not able to learn to classify *useful* and *very useful* pairs. Thus in subtask 2A the test results we obtained were quite poor, while for subtask 2B (see Table 12) we indeed achieved a high Accuracy value (0.79) for two-class classification but a zero Precision value, indicating that the classifier is predicting only the negative class.

3.2 Task 2C – Finding Useful Passages

For this subtask the dataset consisted of each claim text paired with a paragraph that was linked to it. Again the set over which the results could be measured was too small to compare the different parameter settings for the model. In this case the scores obtained without using any external data were quite promising and Table 11 shows the performance versus the number of epochs used for training.

The results for Task 2C in Table 12 show scores that are much lower than the ones obtain in Table 11, nonetheless this submission got the best scores among the teams over Precision (0.41), Recall (0.94) and F1 (0.56), while obtaining a slightly lower result for Accuracy (0.51).

3.3 Task 2D – Assessing Claim Veracity

Subtask D has been tackled thinking about how external data might be leveraged to learn a model for assessing claim factuality. Two different datasets have been

Table 12: Scores for TheEarthIsFlat's official submissions for subtasks 2B and 2C.

Subtask	External Data	Evaluation Method	Precision	Recall	F1	Accuracy
2B (run1)	No	2 Classes per claims	0	0	0	**0.79**
		2 Classes over claims	0	0	0	0.78
		4 Classes over claims	0.28	0.36	0.31	0.59
2B (run2)	FakeNewsChallenge	2 Classes per claims	0	0	0	**0.79**
		2 Classes over claims	0	0	0	0.78
		4 Classes over claims	0.27	0.35	0.3	0.6
2C (run1)	No	2 Classes per claims	0.35	0.72	0.42	0.52
		2 Classes over claims	**0.41**	0.87	**0.55**	0.53
2C (run2)	No	2 Classes per claims	0.4	0.87	0.49	0.51
		2 Classes over claims	0.4	**0.94**	0.56	0.51

considered: The first was the Stanford Natural Language Inference Corpus, [2] while the second was again the FakeNewsChallenge [7] stance detection dataset.

The two datasets have been used to judge the relationship between the claims and the text that composed the web pages. While in the first case the entailment or contradiction confidence score is used, in the second case the confidence over the labels *agree* or *disagree* (how much a text agrees or disagrees with a given headline) was used instead.

The results obtained have been evaluated only over a subset of 31 claims and in this case the best Accuracy value obtained is 0.52.

4 Conclusions

In this report we have described our investigations in applying state-of-the-art pre-trained deep learning models to the problems of *automated claim detection* and *fact checking*, as part of the CLEF'19 Lab: *CheckThat!: Automatic Identification and Verification of Claims*.

For Task A we investigated the use of pre-trained deep neural embeddings for the problem of *check-worthiness* prediction. Over a set of embeddings, we found the Universal Sentence Encoder (USE) [3] to provide the best performance with little out-of-the-box tuning required. We investigated different techniques for pre-processing the political debate data and also the use of external datasets for augmenting the small and highly unbalanced training dataset, but did not observe performance improvements in either case. Thus our runs for the challenge were built by simply training a Feed-Forward neural network on top of the USE encoding(s), without further modification of the training data.

The results obtained for the first task were quite inspiring. With a more judicial choice of validation set it may have been possible to determine that the best choice of model was indeed that used for our third run, which obtained the highest MAP value over all teams for the task. Further work should be aimed at levelling the differences in performance over the different debates.

The various subtasks of Task 2 involved predicting the usefulness of webpages and passages for determining the veracity of a particular claim as well as predicting the veracity of the claim itself. For this task we made use of the BERT [5] model, which can be trained on text pairs to directly predict a relationship label. We found this approach to the task promising, but hampered by insufficient training data and large memory requirements for the BERT model. Furthermore, we found that external datasets (from the FakeNewsChallenge [7]) may be useful for improving performance on these tasks, despite the fact that they are in a different language (English) from the training/test data for the task (Arabic).

In conclusion, the preliminary results show that pre-trained deep learning models can be effective for a variety of tasks. The use of small or unbalanced datasets is a renown problem for deep learning, yet the transfer learning techniques that we used to face the challenge proved quite successful and may offer an opportunity in overcoming deep learning limitations.

References

1. Atanasova, P., Nakov, P., Karadzhov, G., Mohtarami, M., Da San Martino, G.: Overview of the CLEF-2019 CheckThat! Lab on Automatic Identification and Verification of Claims. Task 1: Check-Worthiness
2. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics (2015)
3. Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R.S., Constant, N., Guajardo-Cespedes, M., Yuan, S., Tar, C., Sung, Y., Strope, B., Kurzweil, R.: Universal sentence encoder. CoRR **abs/1803.11175** (2018)
4. Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A.: Supervised learning of universal sentence representations from natural language inference data. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. pp. 670–680. ACL, Copenhagen, Denmark (September 2017)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Elsayed, T., Nakov, P., Barrón-Cedeño, A., Hasanain, M., Suwaileh, R., Da San Martino, G., Atanasova, P.: Overview of the CLEF-2019 CheckThat!: Automatic Identification and Verification of Claims. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. LNCS, Lugano, Switzerland (September 2019)
7. FakeNewsChallenge organizers: FakeNewsChallenge stance detection dataset. http://www.fakenewschallenge.org (2016), online; Since December 1st 2016
8. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Networks **18**(5-6), 602–610 (2005)
9. Hasanain, M., Suwaileh, R., Elsayed, T., Barrón-Cedeño, A., Nakov, P.: Overview of the CLEF-2019 CheckThat! Lab on Automatic Identification and Verification of Claims. Task 2: Evidence and Factuality

10. Iyyer, M., Manjunatha, V., Boyd-Graber, J.L., III, H.D.: Deep unordered composition rivals syntactic methods for text classification. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers. pp. 1681–1691 (2015)
11. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
12. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
13. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 807–814 (2010)
14. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: In EMNLP (2014)
15. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proc. of NAACL (2018)
16. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 5998–6008. Curran Associates, Inc. (2017), http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf
17. Wang, W.Y.: "liar, liar pants on fire": A new benchmark dataset for fake news detection. CoRR **abs/1705.00648** (2017)