DOMLIN at SemEval-2019 Task 8: Automated Fact Checking exploiting Ratings in Community Question Answering Forums

Dominik Stammbach
DFKI, Saarbrücken, Germany
dominik.stammbach@dfki.de

Stalin Varanasi
DFKI, Saarbrücken, Germany
stalin.varanasi@dfki.de

Günter Neumann
DFKI, Saarbrücken, Germany
neumann@dfki.de

Abstract

In the following, we describe our system developed for the Semeval2019 Task 8. We fine-tuned a BERT checkpoint on the qatar living forum dump and used this checkpoint to train a number of models. Our hand-in for subtask A consists of a fine-tuned classifier from this BERT checkpoint. For subtask B, we first have a classifier deciding whether a comment is factual or non-factual. If it is factual, we retrieve intra-forum evidence and using this evidence, have a classifier deciding the comment's veracity. We trained this classifier on ratings which we crawled from qatarliving.com.

1 Introduction

This paper contains our system description for the SemEval2019 task 8 about Fact Checking in Community Forums. The task 8 is divided into two subtasks: In subtask A, the goal is to determine whether a question asks for a factual answer, an opinion or is just posed to socialize. In subtask B, if we have a question asking for a factual answer, we classify the answers to such a question into three categories, namely the answer is either true, false or non-factual, i.e. it does not answer the question in a factual way.

For subtask A, we trained a BERT classifier on the training set and optimized hyper-parameters on the development set. For subtask B, we decided to tackle the challenge with two binary classifiers: Firstly, we decide whether a comment is factual or not. If our classifier decides that a comment is factual, we retrieve intra-forum evidence to determine the comment's veracity using a textual entailment approach. Given the small training set for subtask B, we decided to leverage openly available information on qatarliving.com to create a medium-sized training set. We found that comments on qatarliving.com are sometimes associated with ratings[1] (ranging from 1 to 5) and discovered that high ratings often correspond to replies answering the question in a true way. If a comment has recieved a low rating, we inferred that the comment was most likely not helpful to answer the question and therefore we decided to treat it as a false reply.

2 Related Work

Automated Fact Checking is recently mostly perceived as a number of tasks which can be pipelined together. In the FEVER shared task, most participating systems would first find evidence and then train textual entailment models (Thorne et al., 2018). Related work for Fact Checking in community forums considers a multi-faceted approach incorporating firstly what is said, how it is said and by whom and secondly external evidence from either the web or from the forum itself (Mihaylova et al., 2018). An SVM is trained on top of these features to decide the veracity of a comment.

In our system, we took a similar approach by first retrieving possible evidence, secondly filtering such evidence (through another classifier) and eventually train a system which decides the veracity of a comment based on whether the comment is entailed by the found evidence or not.

3 System Description

Recent progress in natural language understanding shows that pre-training transformer decoders on language modelling tasks leads to remarkable transferable knowledge which boosts performance on a wide range of NLP tasks (Radford et al., 2018). The most recent development then is the

[1] We learnt after the deadline of the shared task that these ratings were automatically generated: https://www.qatarliving.com/forum/technology-internet/posts/searching-information-qatar-living-has-just-grown-faster

Deep Bidirectional Transformers (BERT) which is jointly pre-trained on a masked language modelling task (therefore bidirectional) and on a next-sentence prediction task pushing already impressive results even further (Devlin et al., 2018). All our classifiers in our hand-in are fine-tuned BERT models.

3.1 Domain Adaptation

We firstly fine-tuned a BERT checkpoint (pre-trained on uncased English data only) on the unannotated dataset from Qatar Living with 189,941 questions and 1,894,456 comments (Nakov et al., 2016). Fine-tuning a BERT checkpoint on a new domain consists of further training it jointly on the masked language modelling task and the next-sentence prediction task. For this dataset, it is not always trivial to decide what a sentence is and we use whole comments later on anyways, so we replaced the next-sentence prediction task by a next-comment prediction task, that is our model has to guess whether two comments are appearing consecutively in a thread or not.

Given the peculiarities of the BERT tokenizer, we cleaned the dataset through the following steps:

- we lowercased all characters
- we replaced a character which appears more than three times consecutively to only appear three times (”!!!!!!!!!!” then becomes ”!!!”)
- we removed user specific quotes
- we removed comments containing a type/token ratio[2]of less than 0.15 (because we noticed that they are mostly spam)
- we replaced urls with a special token ”url”, phone numbers with a special token ”tel” and email addresses with a special token ”email”

In Table 1, we show the masked language modelling accuracy (MLM) and next-comment prediction accuracy (NC) for the uncleaned and the cleaned version, both fine-tuned for 100k steps. We also show results for training a task-specific model for subtask A (accuracy on the development set) with the stand-alone BERT model, a fine-tuned model on the raw data and a fine-tuned model on the cleaned data.

[2]https://en.wikipedia.org/wiki/Lexical_density

System	MLM	NC	task A
not fine-tuned	-	-	0.80
fine-tuned raw data	0.68	1	0.79
fine-tuned cleaned data	0.57	0.89	**0.84**

Table 1: Effect of cleaning the dataset

We capped characters to only appear maximum three times consecutively. If they appear more often, they would form a subword anyways and we think it is too easy for the model to guess such subwords in longer sequences (consider the sequence ”!!!!<MASKED>!!!!!”). Users in the forum can add specific quotes which are appended to their posts, e.g. one user chose the ending *life's too short so make the most of it; you only live but once...* which appears 3865 times in the data. We refer to this as ”user specific quotes” and removed them as we believe the model would overfit on such quotes during fine-tuning and would not learn useful knowledge about the domain while doing so. Lastly, we believe that there is not much value to be gained in learning urls, phone numbers and emails, and they often get splitted into a long series of subword units (the vocabulary is managed through byte-pair encoding). We think, these reasons combined make the model learn such patterns very well (resulting in a higher accuracy for the BERT tasks for the model trained on the raw data), but it does not gain much transferable knowledge by doing so, resulting in a lower accuracy for subtask A.

3.2 Subtask A

For subtask A, we trained a task-specific BERT classifier from the fine-tuned BERT checkpoint explained above. Fine-tuning such a classifier consists of learning embeddings for a special classification token, let the model compute self-attention over its 12 layers and finally gather the hidden representation of the classification token (the first token in the sequence usually). This hidden representation is fed into one hidden layer and lastly one classification layer. The input to the model is the concatenation of the question's subject and its body and we regularize the model by applying a dropout of 0.1 on the classification layer. We grid-searched over the proposed hyper-parameter range in the BERT paper (that is initial learning rate, batch-size and number of fine-tuning epochs) (Devlin et al., 2018).

In Table 2, we report the accuracy on the development set for a number of experiments with different features. RelQBody (the opening post by the thread creator) is the question's body, RelQSubject the question's subject (the title of a thread) and RelQ_Category its category (the name of the sub-board it has been posted in). We concatenated the different features with whitespaces in between.

Feature	acc
RelQBody	0.82
RelQSubject + RelQBody	**0.84**
RelQ_Category + RelQSubject + RelQBody	0.83

Table 2: Accuracy for different features for subtask A

Using only the question's body results in slightly worse results than the concatenated subject and body. We also tried to add the category, that is the name of the sub-forum a question has be posted in. The rationale here is that one subboard on qatarliving is called "Socialising" and we thought it might give the model a cue that questions there are more prone to be of the class socializing. However, we get slightly worse results by including it. Our final hand-in eventually consists of an ensemble of 5 models (the voting strategy is majority voting) which are trained on the concatenation of the subject and the body of a question.

Our system ranked fifth with an accuracy of 82% on the test set.

3.3 Subtask B: Overview

As we described earlier, we decided to tackle subtask B as a series of different tasks and for each, we trained different models:

1. decide whether a comment is factual or non-factual
2. retrieve related threads (based on the question of a thread)
3. filter for relevant comments in related threads
4. train a textual entailment[3] system, that is whether the evidence entails a claim or not

For the first step, we have fine-tuned a BERT checkpoint on the SQuAD question answering corpus (Rajpurkar et al., 2016). If a comment contains the answer to a question, we consider it as factual and have to check its veracity in a further step. If the answer to a question can not be found in the comment, we label it as non-factual. If the answer can be found in a comment, i.e. we have a factual comment, we continue with steps 2-4.

For the second step, we search for intra-forum evidence in the qatar living forum dump (Nakov et al., 2016). We concatenate the subject and body of each thread. We lowercase all the tokens, remove all characters except the letters a-z and use the snowball stemmer (Porter, 2001) for stemming the tokens. Afterwards, we search for the most similar threads using TF-IDF[4] and keep the five most similar threads.

We also manually evaluated whether gigablast[5] and the duckduckgo API[6] would yield useful evidence, but after having checked 15 sampled questions from the development set manually, we decided to not pursue this any further. First of all, if we just use the question's subject concatenated with its body as the query for the search engine, it would not be precise and most such queries would not return relevant web pages. One has to summarize this large text of the question automatically into a query suitable for a web search engine. We manually created search-engine searchable queries for the 15 sampled questions and found that only two of such queries returned relevant results. This may be because there is less information available on the internet for queries regarding living in Qatar except for the forum qatarliving.com itself. Hence, we decided to let go of the idea of using publicly available web search engines with automatically summarized questions for this task.

For the third step, we trained a BERT model on the concatenation of the SemEval2016 task 3 subtask A and subtask C data to filter the intra-forum evidence. The input to the model is the original question (the one we want to fact-check comments for) and the found replies in the most similar threads. The output is whether a comment answers that question in a relevant way (yes or no). For the test set for task B, we found 642 comments via the TF-IDF search engine and after filtering the comments, we are left with 162 comments as evidence (24% of these 642 comments).

For the fourth and last step, we also used a BERT model. This model should predict the veracity of a comment given the retrieved evidence in step two and three. However, given the small

[3]https://en.wikipedia.org/wiki/Textual_entailment

[4]https://radimrehurek.com/gensim/

[5]https://www.gigablast.com/

[6]https://duckduckgo.com/api

size of the training set for subtask B (135 false and 166 true comments), we did not manage to find a suitable hyper-parameter configuration which would yield a model with decent performance on the development set.

3.4 Subtask B: Textual Entailment Model

While looking at the forum online, we noticed that some comments in the forum are associated with ratings (Figure 1). Such ratings can range from 1 to 5 and we found that comments with a rating of 5 tend to answer questions in a true way and comments with a rating of 2 or 3 tend to have not been that helpful (we did not find any comments with a rating of 1).

Hence, we have crawled the threads from the forum dump (Nakov et al., 2016) online so that we get the corresponding ratings. We found that the url of a thread is a combination of the sub-forum a thread has been posted in and its subject (with whitespaces replaced with a "+" and some stopwords removed) and reverse engineered the name of the urls. We ignored the threads for which we couldn't find the corresponding web page automatically. After having crawled the website for one night (with short pauses after each call to the website), we ended up with 19'000 comments with a rating of 5 and 13'000 comments with a rating of 2 or 3, resulting in a corpus with 32'000 examples. With this corpus, we trained a textual entailment system which predicts whether a comment is associated with a rating of 2-3 or 5 (we left out comments with a rating of 4 and comments without a rating).

We then retrieved intra-forum evidence as described above for all these 32'000 comments and trained our BERT checkpoint (which was pretrained on the forum dump) on that corpus and obtain "question-comment-evidence" triplets. Let us assume the question is *"Where can I get Potassium Nitrate?"*, the comment is *"Try Metco industrial area. 465 1234"* and we retrieve two evidence texts *"potassium nitrate are not allowed to buy here in qatar. you have to ask a permission from the police department or to the civil defense..."* and *"not sure if same as what you want; but i got potassium before from pharmacies..."*. We then form two triplets (one for each evidence text) and let the model predict an output for each.

Since the different retrieved evidence for each claim is independent, we thought that it would be a bad idea to just concatenate all the evidence and use that as input to our classifier. We therefore decided to aggregate the outputs of each triple using the logsumexp function (Eq. 1) which is a smooth version of the max function and allows the model to back-propagate dense gradients (Verga et al., 2018). We think this lets the model also figure out on its own which evidence it should look out for.

$$scores(i) = log \sum exp(A_{ij}) \quad (1)$$

A is a matrix with two columns (bad rating or good rating) in which we stack the predictions for each "question-comment-evidence" triplet. That is, each row in that matrix is the prediction for a comment with a rating given one evidence comment found in the forum. In comparison to the normal max function (which back-propagates sparse gradients), we learn from each comment-evidence pair and not only from the one with the highest scores.

We trained that model with a batch-size of 8 answers and for each answer, we retrieve 4 evidence comments (resulting in 32 triplets). During test time, we retrieve up to 8 evidence comments, predict results for each triplet and aggregate the predictions for each triplet using the logsumexp function to yield a final classification. In Table 3, we show the results of our two classifiers on the training set of subtask B (because we did not use that set for training at all).

class	pr	rc	F1
non-factual	0.43	0.53	0.47
factual	0.63	0.53	0.58
factual false	0.36	0.52	0.42
factual true	0.71	0.56	0.62

Table 3: Results on training set of subtask B

The first two rows show the results of our BERT model trained on the SQuAD corpus. The factual class contains the examples which are true or false. After having performed a manual error analysis for the factual and non-factual class, we conclude that we disagree with some of the annotations in the training corpus. The last two rows show the performance of our classifier trained on ratings on the training set. For the true answers, it performs better than for the false answers (which might be due to a slight imbalance of training examples in our compiled corpus).

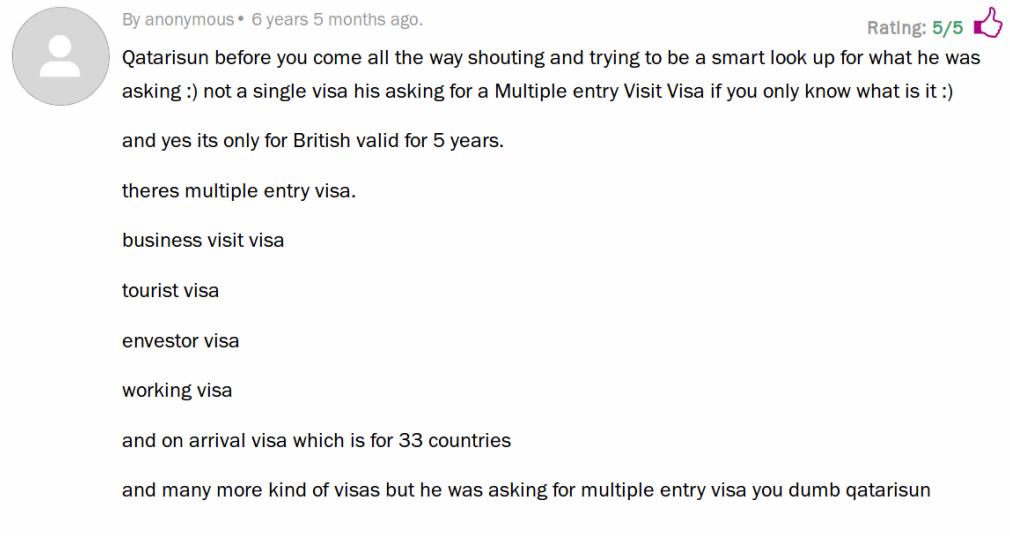

Figure 1: Comment with an associated Rating

3.5 Subtask B: Contrastive Runs

We only handed in contrastive runs for subtask B. The difference to our original hand-in is solely the classifier deciding whether a comment is factual or non-factual. In our first contrastive run, we used the BERT model pre-trained on the concatenation of the SemEval2016 task 3 subtask A and C data (the same we use to filter evidence). For our second run, we used a ranking model to get a similarity score between a question and a comment based on the ratings. We minimized the following

$$\text{loss} = \sum_i max(0, \delta - cos(q_i, comment_{5i}) + cos(q_i, comment_{0i}))$$

where i is a data point from the web-crawled corpus, $comment_{5i}$ is the vector obtained by the model for a comment with rating 5, $comment_{0i}$ is the obtained vector by the model for a comment without a rating, q_i is the model obtained vector for the corresponding question, δ(=0.1) is the allowed margin between a positive similarity and a negative similarity which is chosen as a hyperparameter. All vectors are obtained by max pooling the hidden states of an encoder BI-LSTM on the input text (question/comment). Our assumption is that the comment with rating 5 will be a factual answer in most of the cases (noisily labelled). Furthermore, we fine-tuned this model for the answer classification task on the training dataset for the labels 'non-factual' and 'true/false'. In table 4, we report the results for our different runs on the test set.

We also submitted an all non-factual baseline

run	**acc** (%)	F1	AvgRec	MAP
main	**0.72**	0.4	0.44	0.27
1. Contrastive run	**0.81**	0.48	0.53	0.21
2. Contrastive run	**0.48**	0.21	0.31	0.29
non-factual baseline	**0.83**	0.28	0.33	0.29

Table 4: Results of different runs for subtask B on test-set

on the test set and it scored 83% accuracy. We think this biased test set hence does not reflect the model's ability to fact check comments. We reckon that in further work on this dataset, one should therefore not focus on accuracy but on a different metric.

4 Conclusion

We described our hand-in for the semeval2019 task 8. For subtask A, we fine-tuned a BERT checkpoint pretrained on a cleaned qatar living forum dump. For subtask B, we decided to use two classifiers. One classifier decides whether a comment is factual or non-factual. If it is factual, a second classifier makes a prediction about the comment's veracity. Given the small size of the training dataset, we crawled qatarliving.com to generate a medium sized, weakly supervised training corpus based on ratings in the forum. To train our model, we searched for intra-forum evidence for every comment and fine-tuned a BERT classifier for each question-comment-evidence triplet. Since the retrieved evidence is independent of each other, we did not concatenate all the evidence for a question but aggregated results for each triplet using the logsumexp function. We de-

cided to use this function for aggregation because it allows the model to send back dense gradients and learn from all the comment-evidence pairs and not only the evidence with the highest score.

5 Acknowledgements

This work was partially supported by the German Federal Ministry of Education and Research (BMBF) through the project DEEPLEE (01IW17001).

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tsvetomila Mihaylova, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, Mitra Mohtarami, Georgi Karadzhov, and James R. Glass. 2018. Fact checking in community forums. In *AAAI*, pages 5309–5316. AAAI Press.

Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, abed Alhakim Freihat, Jim Glass, and Bilal Randeree. 2016. Semeval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 525–545. Association for Computational Linguistics.

Martin F. Porter. 2001. Snowball: A language for stemming algorithms. Published online. Accessed 11.03.2008, 15.00h.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and verification (fever) shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 1–9. Association for Computational Linguistics.

Patrick Verga, Emma Strubell, and Andrew McCallum. 2018. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 872–884. Association for Computational Linguistics.