HHU at SemEval-2018 Task 12: Analyzing an Ensemble-based Deep Learning Approach for the Argument Mining Task of Choosing the Correct Warrant

Matthias Liebeck, Andreas Funke, Stefan Conrad
Institute of Computer Science
Heinrich Heine University Düsseldorf
D-40225 Düsseldorf, Germany
{liebeck,conrad}@cs.uni-duesseldorf.de
andreas.funke@hhu.de

Abstract

This paper describes our participation in the SemEval-2018 Task 12 *Argument Reasoning Comprehension Task* which calls to develop systems that, given a reason and a claim, predict the correct warrant from two opposing options. We decided to use a deep learning architecture and combined 623 models with different hyperparameters into an ensemble. Our extensive analysis of our architecture and ensemble reveals that the decision to use an ensemble was suboptimal. Additionally, we benchmark a support vector machine as a baseline. Furthermore, we experimented with an alternative data split and achieved more stable results.

1 Introduction

Argument mining is a trending research domain that focuses on the extraction of arguments and their relations from text. It has been applied to multiple languages and multiple application domains, including the legal domain (Palau and Moens, 2009), persuasive essays (Stab and Gurevych, 2014), online participation (Liebeck et al., 2016), and web discourse (Habernal and Gurevych, 2017). The three most common subtasks are: argument identification, argument classification, and argument linking. These focus on the identification of argumentative text content, the extraction of argument components according to a specific argument model, and the extraction of relations between arguments components, respectively.

Currently, different argument models comprising different argument components are being used throughout the community, such as the *claim-premise family* or Toulmin's model (Toulmin, 1958). In the scope of this paper, claims, premises, and warrants are the most important argument components. Claims are often defined as controversial statements that are either true or false. Premises are reasons that support or attack claims. In Toulmin's model, they are connected with warrants that state why the premise supports the claim.

Habernal et al. (2018b) introduced a new argument mining task called *argument reasoning comprehension* with the following definition: Given a reason and a claim, identify the correct warrant from two opposing options.

This paper describes our participation in the SemEval-2018 Task 12 *The Argument Reasoning Comprehension Task* (Habernal et al., 2018a) that uses the dataset from Habernal et al. (2018b) as a shared task. Besides a description of our machine learning systems, we evaluate additional machine learning models (we were only allowed to submit a single set of predictions for the official ranking) and we further analyze the test set.

The dataset for the challenge consists of annotated news comments from the New York Times user debate section. With Amazon Mechanical Turk as a crowdsourcing platform, 5000 randomly selected user comments were annotated in a multi-step annotation process that included three free text annotation steps (gist summarization, the creation of warrants, and of alternative warants). After the final filtering, the dataset for the Argument Reasoning Comprehesion Task comprises 1970 instances, where each instance is a tuple of (R, C, W, AW, G, T, I) comprising a reason (R), a claim (C), a warrant (W), an alternative warrant (AW), a gold label (G) indicating which of both warrants is correct, a debate title (T), and additional debate information (I) about the debate. The task organizers split the dataset into three distinct groups: training set (1,210 instances), development set (316 instances), and a test set (444 instances).

Figure 1 shows a training example of the dataset. The machine learning task of predicting the correct warrant is difficult since both warrants

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 1114–1119
New Orleans, Louisiana, June 5–6, 2018.

Title: Do We Still Need Libraries?
Debate information: Do We Still Need Libraries? What are libraries for, and how should they evolve?
Claim: We need libraries
Reason: Libraries have lots to offer in addition to books they provide music, dvd's, magazines and more.
✓ **Warrant 1**: all these things are readily available to everyone online
✗ **Warrant 2**: none of these things are readily available to everyone online

Figure 1: Example of a training instance with warrant 1 as the correct gold label.

are lexically similar, and can differ in just one or two words.

In the trial phase of the challenge, the participants were given access to the training set and the dev test with gold labels. In the test phase of the challenge, the task participants were given access to the test set (with omitted gold labels) and had to submit predictions for all test instances.

2 Our Approach

For our participation in the challenge, we experimented with deep learning architectures in Keras (Chollet, 2015) with TensorFlow (Abadi et al., 2015) as the backend. We tested multiple deep learning architectures comprising different layers and input channels. For each of these models, we performed an extensive grid search for hyperparameters, such as layer sizes, embedding sizes, activation functions, optimizers, loss functions, batch sizes, dropout, number of training iterations, and different seeds for the initialization.

In our final experiments, we evaluated four different embeddings: pre-trained fastText embeddings (Bojanowski et al., 2017) on the entire Wikipedia corpus, two word embeddings with different dimensionality trained on the task's dataset with the word2vec (Mikolov et al., 2013) skip-gram model implemented in gensim (Řehůřek and Sojka, 2010), and a fourth embedding based on the task's vocabulary and corresponding Wikipedia articles.

We benchmarked each trained model on the development set. This yielded thousands of trained models. Ultimately, we selected a deep learning architecture with high accuracy scores and a low variance, as outlined below. Our motivation was to select a model that we believed to be stable and able to generalize well on the test set.

2.1 Archictecture

We now outline our deep learning architecture, as visualized in Figure 2. We use warrants, reasons, claims, and alternative warrants as input channels of our neural network. The preprocessing for each channel consists of tokenization, padding, and word embeddings as representations for individual words, as it is common for recurrent neural networks in NLP. First, each input sequence is fed into a bidirectional LSTM (Schuster and Paliwal, 1997; Graves and Schmidhuber, 2005). In the next layer, we use two parallel LSTMs (Hochreiter and Schmidhuber, 1997). The first LSTM uses a concatenation of the warrant, the reason, and the claim as the input, whereas the concatenation of the alternative warrant, reason, and claim is used as the input of the other LSTM. The output of both LSTMs is then concatenated, dropout is applied, and finally mapped as output through two dense layers.

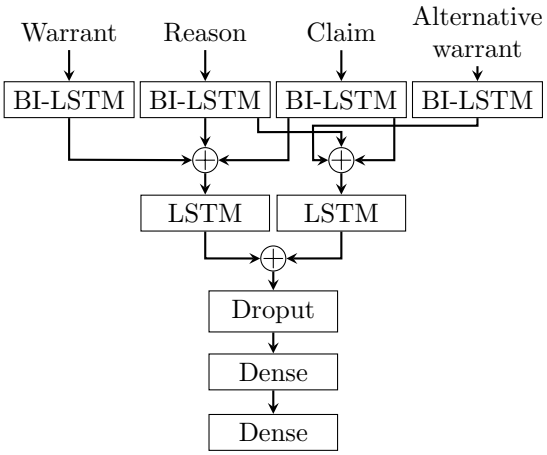


Figure 2: Illustration of our deep learning architecture.

In our experiments, we benchmarked our model with 10 different seeds and achieved an average accuracy of 67.7% (± 2.2%) on the development set.

2.2 Ensemble

We experimented with additional ways of increasing our performance on the development set under the assumption that the test set would be very similar. By using an ensemble of multiple trained models with a majority vote as the prediction, we were

able to improve our results considerably. First, we trained 2560 models of the above-described architecture with four different embeddings and various hyperparameters. Then, we combined all 623 of these models with a development set accuracy of above 67% into our final ensemble. Our ensemble yielded a promising result of 73.3% accuracy on the development set, which is a higher accuracy than all submitted and benchmarked systems in the trial phase of the challenge. Therefore, we decided to use the ensemble for our predictions instead of a single model.

3 Results

We now report the official results of our ensemble on the test set, as well as benchmarks of the single models. Additionally, we compare our deep learning approaches with a support vector machine as a classical machine learning baseline.

3.1 Deep Learning

Our ensemble achieved 17th place in the competition and yielded an accuracy score of 53.4%, which was lower than we anticipated based on our good performance on the development set.

Since we were curious to see whether the decision to use an ensemble was beneficial and in order to better understand the low results on the test set, we further analyzed all trained models on the test set after the release of the gold labels. The performance difference in terms of accuracy scores on the development set and the test set of all 2560 models that we considered for the ensemble is visualized in Figure 3. It can immediately be seen that all our models achieved better scores on the development set than on the test set and that some hyperparameters lead to models yielding bad performances. The most interesting insight from this plot is that the majority of our single models performed better than the ensemble score of 53.4%. Upon further analysis of the 623 models with a development set accuracy of above 67% that we used for our ensemble, we can observe an average accuracy score of 54.4% ($\pm$ 2.0%) on the test set. This also shows that the decision to opt for an ensemble was disadvantageous, since the ensemble was not able to generalize better than individual models.

The influence of the number of selected models from the 2560 available models is further visualized in Figure 4a, where the models were being added in descending order based on the development set performance. In our submission, we decided to use the 623 best-ranked models with a majority voting. Figure 4a shows that an ensemble with a considerably smaller number of models would have performed better on both sets, as the test scores with the majority voting began to deteriorate quickly.

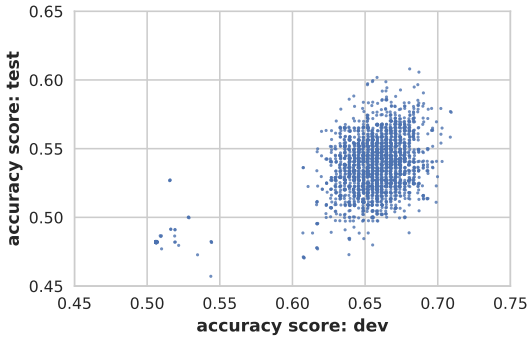


Figure 3: Performances of 2560 trained models with different hyperparameters on the development set and the test set

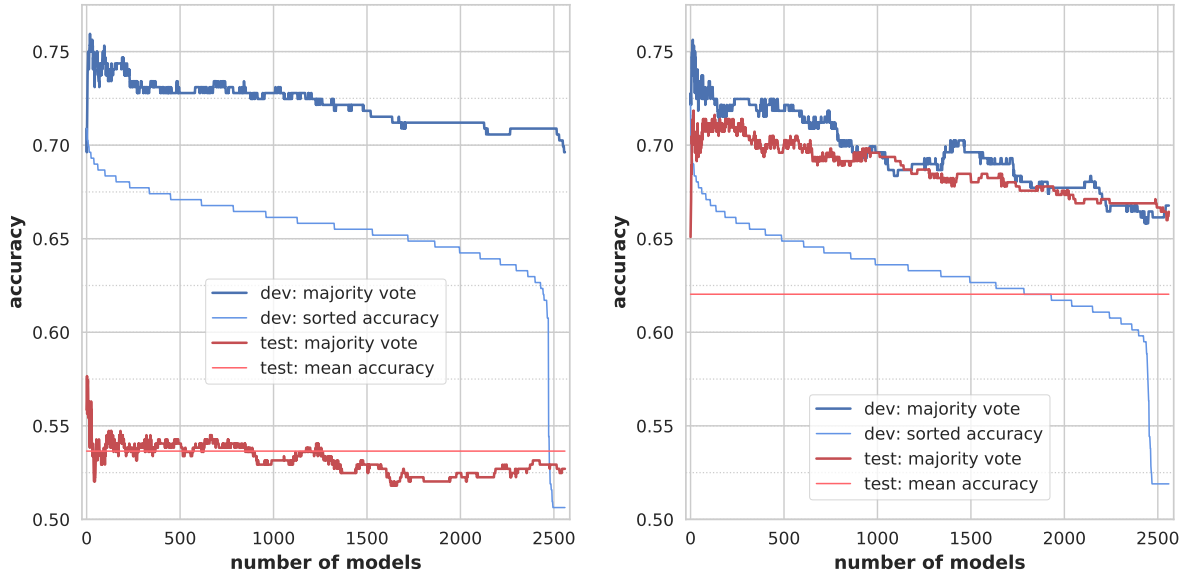
3.2 Support Vector Machine Baseline

Furthermore, we compared our deep learning approaches with a support vector machine (SVM). We used the same input data for the SVM as for our neural networks by using claims, reasons, warrants, and alternative warrants. All four input strings were tokenized, padded to a fixed length, represented by embedding vectors, and concatenated to achieve a fixed input length for the SVM.

In total, we trained 72 SVMs with different hyperparameters. Their performances on both sets are visualized in Figure 5. Compared with our deep learning approach, the results of the SVMs on the development set are lower, with an average score of 58.8% ($\pm$ 2.8%), but comparable for the test set, with 53.6% ($\pm$ 2.3%). Again, we observed lower accuracies for the test set than for the development set.

4 Observations of an Alternative Data Split

The performance difference on both sets motivated us repeat our experiments on an alternative data split, in which we shuffled all data points together and created three new sets (training, development, and test) with the same sizes as the original split.



(a) Original dataset

(b) Ensemble experiment with our alternative data split

Figure 4: Influence of the number of models in our ensemble.

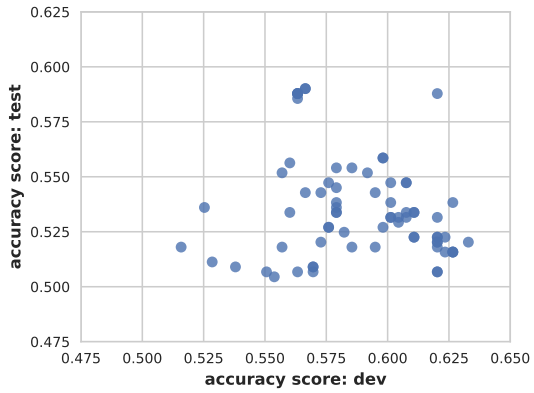


Figure 5: Performances of support vector machines on the development set and the test set.

4.1 Single Model

On the alternative split, we benchmarked our single model with 20 different seeds and achieved higher results of 63.7% (± 1.6%) on the dev set and 62.1% (± 2.5%) on the test set.

4.2 Ensemble

For the ensemble, we trained 2560 models with the same hyperparameters as for the original dataset. The models behaved more similarly on both the alternative development set (63% (± 3.0%)) and the test set (62% (± 4.1%)), as visualized in Figure 6. If we take a look at the performance of the ensemble's majority vote in Figure 4b and compare it with the original dataset in Figure 4a, we can see that the idea of using an ensemble can be beneficial. However, this is dependent on a more evenly represented data split. In hindsight - with posterior knowledge of the test set - it would have been a better choice to decide for the ensemble's peak performance (original split dev: 76%, test: 55.6%; alternative split dev: 75.6%, test: 71.3%).

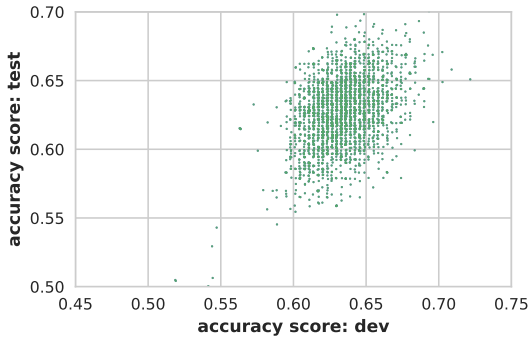


Figure 6: Performances of 2560 trained models with different hyperparameters on the alternative data split.

4.3 Support Vector Machine

We again trained 72 SVMs with different hyperparameters. With the alternative split, we achieved an average score of 55.5% (± 2.3%) on the new dev set and 57.7% (± 2.1%) on the new test set. Compared with our neural network, the SVM approach now performs worse.

5 Conclusion

For our participation in *The Argument Reasoning Comprehension Task*, we benchmarked several deep learning approaches with different layers, embeddings, and hyperparameters. For our final submission, we decided to use an ensemble comprising 623 models with a majority voting that performed much better on the development set than a single model.

Unfortunately, our ensemble underperformed on the test set. Therefore, we extensively analyzed our ensemble approach after the release of the gold labels. We ascertained that the use of an ensemble was a suboptimal choice, and the predictions of most single models would have performed better and that the choices of hyperparameters and seeds influenced the stability of the predictions, as illustrated in Figure 3.

We compared our deep learning approach with a support vector machine as a baseline. Although our models and our ensemble performed much better on the development set, the SVM produced slightly better results on the test set.

Finally, we repeated our experiments on an alternative data split and achieved more stable results. Therefore, we conclude that the test set comprises data points with characteristics that are not present in the original training data.

Because we - and other task participants with deep learning approaches - had trouble providing a satisfying solution to the task, we also believe that additional preprocessing steps are required for a machine learning approach, since the warrants and alternative warrants are so lexically similar.

Acknowledgments

This work was funded by the PhD program *Online Participation*, supported by the North Rhine-Westphalian funding scheme *Fortschrittskollegs*. Computational support and infrastructure were provided by the "Centre for Information and Media Technology" (ZIM) at the University of Düsseldorf (Germany).

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

François Chollet. 2015. Keras. `https://github.com/fchollet/keras`.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Ivan Habernal and Iryna Gurevych. 2017. Argumentation Mining in User-Generated Web Discourse. *Computational Linguistics*, 43(1):125–179.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018a. SemEval-2018 Task 12: The Argument Reasoning Comprehension Task. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, page (to appear). Association for Computational Linguistics.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018b. The Argument Reasoning Comprehension Task: Identification and Reconstruction of Implicit Warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, page (to appear). Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780.

Matthias Liebeck, Katharina Esau, and Stefan Conrad. 2016. What to Do with an Airport? Mining Arguments in the German Online Participation Project Tempelhofer Feld. In *Proceedings of the Third Workshop on Argument Mining (ArgMining2016)*, pages 144–153. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781.

Raquel Palau and Marie-Francine Moens. 2009. Argumentation Mining: The Detection, Classification and Structure of Arguments in Text. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law, ICAIL '09*, pages 98–107. ACM.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50. ELRA.

Mike Schuster and Kuldip Paliwal. 1997. Bidirectional Recurrent Neural Networks. *IEEE Trans. Signal Processing*, 45(11):2673–2681.

Christian Stab and Iryna Gurevych. 2014. Identifying Argumentative Discourse Structures in Persuasive Essays. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 46–56. Association for Computational Linguistics.

Stephen Toulmin. 1958. *The Uses of Argument*. Cambridge University Press.