Embarrassingly Simple Performance Prediction for Abductive Natural Language Inference

Emīls Kadiķis, Vaibhav Srivastav, and **Roman Klinger**
Institut für Maschinelle Sprachverarbeitung
University of Stuttgart
Pfaffenwaldring 5b, 70569 Stuttgart
{emils.kadikis,vaibhav.srivastav,klinger@ims.uni-stuttgart.de}

Abstract

The task of abductive natural language inference (αNLI), to decide which hypothesis is the more likely explanation for a set of observations, is a particularly difficult type of NLI. Instead of just determining a causal relationship, it requires common sense to also evaluate how reasonable an explanation is. All recent competitive systems build on top of contextualized representations and make use of transformer architectures for learning an NLI model. When somebody is faced with a particular NLI task, they need to select the best model that is available. This is a time-consuming and resource-intense endeavour. To solve this practical problem, we propose a simple method for predicting the performance without actually fine-tuning the model. We do this by testing how well the pre-trained models perform on the αNLI task when just comparing sentence embeddings with cosine similarity to what the performance that is achieved when training a classifier on top of these embeddings. We show that the accuracy of the cosine similarity approach correlates strongly with the accuracy of the classification approach with a Pearson correlation coefficient of 0.65. Since the similarity computation is orders of magnitude faster to compute on a given dataset (less than a minute vs. hours), our method can lead to significant time savings in the process of model selection.

1 Introduction

Abduction is a type of reasoning that infers an explanation for some observations (Peirce, 1931). It is a particularly challenging type of inference; as opposed to deduction and induction, which derive the conclusion from only the information present in the observations, abduction requires making assumptions about an implicit context beyond just the given observations. Abductive reasoning is therefore at the core of the way humans understand the world and how world knowledge is involved.

Abductive reasoning in the natural language domain has been introduced by Bhagavatula et al. (2020) who defined the abductive natural language inference (αNLI) task. In it, we are given four sentences – two observations o_1 and o_2 and two hypotheses h_1 and h_2, where we know that the sequence of events was $o_1 \rightarrow (h_1|h_2) \rightarrow o_2$. The task then is to decide which of the two hypotheses is the more plausible one.

An example from Bhagavatula et al. (2020) is the following:

o_1 : It was lunchtime and Kat was hungry.
o_2 : Kat and her coworkers enjoyed a nice lunch outside.
h_1 : Kat went to get a salad.
h_2 : Kat decided to take a nap instead of eating.

While it is not inconceivable that someone would decide to take a nap on their lunch break (h_2), given o_2 the first hypothesis becomes more likely.

Currently, transformer-based architectures (Vaswani et al., 2017) are state of the art in a wide variety of natural language processing (NLP) tasks (Devlin et al., 2019; He et al., 2021; Li et al., 2021), including αNLI. However, with an ever-changing landscape of transformer models and pre-training techniques (with over 10000^1 different fine-tuned models available on the HuggingFace hub (Wolf et al., 2020)), finding the best model for a given task has become a time-consuming process since, in order to compare multiple models, they each need to be separately fine-tuned on the task.

This model selection process might lead to a prohibitive runtime, which has led to research on performance prediction, namely to predict the expected performance out of parameters of the model configuration, without actually training the model. This procedure has been evaluated for a set of NLP tasks, including span prediction (Papay et al., 2020) and language modelling (Chen, 2009). However,

---

[1]https://huggingface.co/models

we are not aware of any previous work that performed performance prediction for αNLI.

In this paper, we introduce a fast performance prediction method for the αNLI task that allows a more guided way of choosing which models to fine-tune. We use various pre-trained transformer models to embed the observations and hypotheses with the approach introduced in Reimers and Gurevych (2019), then compare which hypothesis is closer to the observations with cosine similarity. We find that the performance of the similarity-based approach is correlated to results obtained via fine-tuning. Therefore, the similarity-based approach can serve as a performance prediction method.

2 Related work

There are three research topics that need to be mentioned. Approaches to abductive reasoning, pre-trained language models, and performance prediction. In this section, we explore them in detail.

Abductive natural language inference. NLI has been proposed as the task of recognizing textual entailment by MacCartney and Manning (2008) and now constitutes a major challenge in NLP which has found application for other downstream tasks, including question answering or zero-shot classification (Yin et al., 2019; Mishra et al., 2020). Based on the initial goal of establishing inference relations between two short texts, a myriad of variants have been proposed (Yin et al., 2021; Williams et al., 2018; Bowman et al., 2015). One such variant is abductive NLI (αNLI, Bhagavatula et al., 2020).

Transformer-based architectures dominate the αNLI leaderboard,[2] including RoBERTa-based models (Liu et al., 2020; Mitra et al., 2020) which explore how additional knowledge can improve performance on reasoning tasks, and Zhu et al. (2020) who approach αNLI as a ranking task. The task authors improved upon their result in Lourie et al. (2021) by using a T5 model (Raffel et al., 2020) and experimenting with multi-task training and fine-tuning over multiple reasoning tasks. Both the multi-task criteria and the Text-to-Text framework of T5 help the model generalize and understand the context better.

The second-best model on the leaderboard is a DeBERTa model (He et al., 2021). The model re-

places the masked language modelling task with a replaced-token detection task. It also uses a disentangled attention mechanism to encode the position and content information.

The current state of the art shows an accuracy of 91.18% using a new unified-modal pre-training method to leverage multi-modal data for single-modal tasks (Li et al., 2021). This result approaches the human baseline of 92.9%.

Pre-trained language models. The αNLI task requires the model to successfully "understand" the context of both the observations and use that understanding to identify the more likely hypothesis entailing it. Most semantic representations in practical applications rely on distributional semantics. Such methods include the word-level embedding methods Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) and language model-based word representation like ELMo (Peters et al., 2018), ULMFit (Howard and Ruder, 2018), and GPT (Radford et al., 2018).

The current state of the art are pre-trained transformer architectures (Vaswani et al., 2017) like BERT (Devlin et al., 2019), which use a masked language modelling and next sentence prediction objective. This not only helps the model understand the context within a sentence but also in-between consecutive sentences. There is however a trade-off in terms of the time it takes to train transformer models. For example, a from-scratch training of BERT takes 6.4 days on an 8 GPU Nvidia V100 server[3]. Devlin et al. (2019) recommend fine-tuning the language model between 2-4 epochs for a given task. However, in practice, multiple trials are required to find the optimal hyperparameters. These long training times and multiple fine-tuning runs make model selection a time-intensive process (Liu and Wang, 2021).

Performance prediction. The task of performance prediction is to estimate the performance of a specific system without explicitly running it. It helps in setting hyperparameters, finding feature sets, or identifying candidate language models for a downstream NLP task. Chen (2009) develop, for instance, a generalized method for predicting the performance of exponential language models. They analyze the backoff features in an exponential n-gram model. Papay et al. (2020) leverage

---

meta-learning to identify candidate model performance on the task of span identification. They train a linear regressor as a meta-model to predict span ID performance based on model features and task metrics for an unseen task. Ye et al. (2021) propose performance prediction methods particularly suited for fine-grained evaluation metrics. They also develop methods for estimating the reliability of these performance predictions. Contrary to the previously mentioned papers, Xia et al. (2020) build regression models to predict the performance across a variety of NLP tasks, however, they do not consider NLI as one of them.

In contrast to our work, all these previous methods build on top of the idea to train a surrogate model for performance prediction and depend on the information about past runs of these models. Our approach focuses solely on the embeddings provided by the language model and leverages those as a predictor of performance. This particular setup is also motivated by the αNLI task itself, in which a sentence needs to be chosen for a given set of other sentences.

3 Methods

Our paper investigates how well a fine-tuned transformer model's performance on the αNLI task (Bhagavatula et al., 2020) is approximated by the cosine similarity of embeddings of the input sentences which we obtain from the pre-trained models before fine-tuning them.

Intuitively, if a model embeds the correct hypothesis close to the observations in some latent space (not necessarily a semantic similarity space), then a classification model built on top of that latent space should have an easier time discerning which is the correct hypothesis, because apparently that latent space captured some features that were salient for the αNLI task.

3.1 Sentence Embedding

For both the similarity baseline and the fine-tuned classification model, the starting point is the pre-trained transformer model itself. We add a mean pooling layer to convert the token-by-token output of the model into a single sentence embedding (Reimers and Gurevych, 2019). Given some tokenized input $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ and a pre-trained transformer model E which encodes each token $E(x_i)$, we calculate the sentence embedding $\text{emb}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} E(x_i)$.

Al alternative to mean pooling would have been to use the embedding of the CLS token. We opted against that for three reasons. Firstly, Reimers and Gurevych (2019) show that mean pooling slightly outperforms using the CLS token in their semantic similarity models. Secondly, for some models, the CLS token does not have any particular significance before fine-tuning on the downstream task due to the training objective they use (such as RoBERTa (Liu et al., 2020), which uses masked language modelling). Thirdly, pooling is a general approach that can be adopted for any model, even if it does not output a CLS token. Since our goal was to avoid any model-specific enhancements, a universal blanket approach like this was preferable.

3.2 Similarity-based αNLI

To perform αNLI on a validation instance, we obtain three sentence embeddings – one for the combined observations $o_1 + o_2$ and one for each of the hypotheses h_1, h_2. To predict the more plausible hypothesis, we calculate which of them is closer to the observations with cosine similarity:

$$\hat{h} = \underset{h' \in \{h_1, h_2\}}{\arg\max} \cos(\text{emb}(o_1 + o_2), \text{emb}(h'))$$

3.3 Classification-based αNLI

For the classification model, we add a classification head on top of the pre-trained model, which consists of a mean pooling layer to get sentence embeddings, then a fully-connected layer and a softmax output layer. For each instance of (o_1, o_2, h_1, h_2), the model takes two different inputs which consist of both observations with each of the hypotheses, namely $\text{emb}(o_1 + o_2 + h_1)$ and $\text{emb}(o_1 + o_2 + h_2)$.

Both of these input representations are then used in a fully connected layer f with a softmax output layer to get the probability score for each input. The hypothesis that is assigned the largest probability constitutes the prediction:

$$\hat{h}' = \underset{h \in \{h_1, h_2\}}{\arg\max} \text{softmax}(f(\text{emb}(o_1 + o_2 + h)))$$

In our experiments, we fine-tune the classification head on the αNLI training set without updating weights in the underlying language model. This is mostly due to time and resource constraints, however, we believe that while fine-tuning would improve classification performance across the board, it would not affect the ranking as such. Since we are comparing models amongst themselves, the ranking is more important.

		Accuracy		Run time	
Model	Citation	Sim.	Class.	Sim.	Class.
albert-base-v2	Lan (2020)	50.78%	65.34%	5.68	0:55:58
albert-large-v2	Lan (2020)	50.13%	69.71%	7.93	2:51:01
bert-base-uncased	Devlin (2019)	51.69%	65.99%	2.69	1:13:28
bert-large-uncased	Devlin (2019)	52.67%	67.04%	7.22	3:13:54
distilbert-base-uncased	Sanh (2019)	51.63%	62.60%	1.63	0:26:36
squeezebert/squeezebert-uncased	Iandola (2020)	50.52%	61.95%	2.23	0:37:00
google/mobilebert-uncased	Sun (2020)	48.75%	61.68%	4.29	0:36:20
google/canine-s	Clark (2021)	49.21%	58.09%	3.82	1:30:40
google/electra-small-discriminator	Clark (2020)	51.17%	63.51%	1.69	0:12:27
google/electra-base-discriminator	Clark (2020)	52.28%	78.07%	2.77	0:51:51
google/electra-large-discriminator	Clark (2020)	52.74%	88.51%	7.23	3:09:35
microsoft/mpnet-base	Song (2020)	51.50%	74.87%	2.77	0:52:36
roberta-base	Liu (2020)	51.50%	74.15%	2.70	0:52:28
roberta-large	Liu (2020)	51.63%	84.14%	6.87	3:32:47
google/bigbird-roberta-base	Zaheer (2020)	51.50%	71.02%	5.74	1:02:08
kssteven/ibert-roberta-base	Kim (2021)	51.50%	73.63%	2.78	1:00:39
distilroberta-base	Sanh (2019)	51.43%	65.99%	1.67	0:29:08

Table 1: Accuracy on the αNLI validation set using similarity and a classification model. The similarity runtime (how long it takes to evaluate the model since no training is required) is shown in seconds, the classification runtime (how long it takes to fine-tune and evaluate the model) in hours, minutes, and seconds. Note that the given training time is for a single set of hyperparameters. Identifying the best hyperparameters involved training each model multiple times.

4 Experiments

We compare the similarity-prediction-based αNLI approach and the classification-based αNLI approach to evaluate if the first can act as an approximation for the performance expected by the second. We use the pre-trained transformer models which are available on the HuggingFace (Wolf et al., 2020) hub. The full list of models we use is listed in Table 1. The code for the experiments is available online.[4]

4.1 Dataset

All of our experiments were run on the train and validation split in the ART dataset provided for the αNLI challenge (Bhagavatula et al., 2020). It consists of 169,654 training and 1,532 validation samples, each consisting of two observations and two hypotheses obtained from a narrative short story corpus and augmented with wrong hypotheses written by crowd-sourced workers.

The training data contains repetitions of the same (o_1, o_2) pairs with different sets of hypotheses, ranging from one plausible and one implausible hypothesis to two plausible hypotheses where one of them is more plausible. The validation set was constructed using adversarial filtering, which selects one plausible and one implausible hypothesis

for each set of observations that are hard to distinguish. This increases the probability that the instances are free of annotation artifacts, which the authors defined as "unintentional patterns in the data that leak information about the target label" (Bhagavatula et al., 2020).

4.2 Experimental Setup

All of our classification and similarity experiments were run on an Nvidia RTX 2080 GPU. For training the classifier we used the maximum batch size that fit on the GPU (which is different for different sized models, ranging between 12 and 128). For similarity experiments, we only infer the embeddings from pre-trained models. For hyperparameter selection, to keep the comparison fair, we tuned the batch size and learning rate and considered the same set of possible combinations across all the models. The specific values used for each model are available in Table 2 in the appendix. We train for 3 epochs with learning rates ranging $[10^{-5}; 9 \cdot 10^{-5}]$ and a weight decay of 0.01. For each pre-trained model, we pick the one that achieved the highest accuracy on the validation set.

4.3 Evaluation and Results

Table 1 shows the results as accuracy scores, obtained with each pre-trained model when using cosine similarity and when using a classifier built on top of it. We also show the training runtimes.

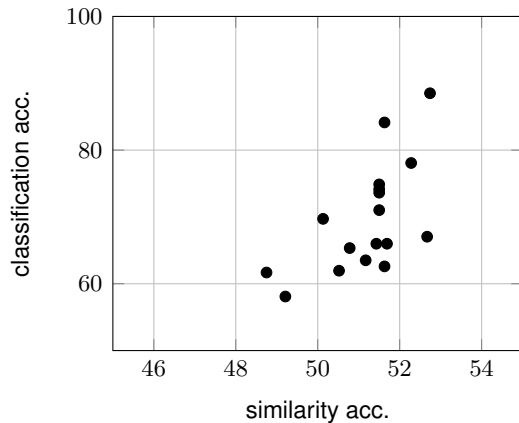[4]https://github.com/Vaibhavs10/anli-performance-prediction

Figure 1: Relation between similarity and classification accuracy scores. Note that the similarities are closer to each other than the classification values, therefore we chose a different scale.

The primary observation is that the accuracy scores of classification and similarity are correlated. This can be seen in Figure 1. The Pearson correlation coefficient is $r = .65$ ($p = 0.005$). The Spearman's correlation coefficient is $\rho = .67$ ($p = 0.003$), indicating that the ranking obtained with the similarity-based prediction is a reliable indicator that is helpful for model selection. Additionally, model fine-tuning takes on average 620 times longer than the similarity-based estimate. Tuning the hyperparameters involved training each model multiple times.

5 Conclusions & Future Work

In this paper, we have shown that similarity measures based on the distributional semantic representation in pre-trained transformer models serve as an effective proxy for fine-tuned transformer-based classification in αNLI. Since fine-tuning a transformer model takes notably more time than performing similarity comparisons, our approach supports efficient model selection procedures. Future work should investigate the suitability of similarity-based performance prediction for other similar tasks, like next sentence prediction, question answering, summarization.

References

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Stanley Chen. 2009. Performance prediction for exponential language models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 450–458, Boulder, Colorado. Association for Computational Linguistics.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2021. CANINE: pre-training an efficient tokenization-free encoder for language representation. *CoRR*, abs/2103.06874.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Forrest Iandola, Albert Shaw, Ravi Krishna, and Kurt Keutzer. 2020. SqueezeBERT: What can computer vision teach NLP about efficient neural networks? In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 124–135, Online. Association for Computational Linguistics.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. I-bert: Integer-only bert quantization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5506–5518. PMLR.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of the International Conference on Learning Representations*.

Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. 2021. UNIMO: Towards unified-modal understanding and generation via cross-modal contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2592–2607, Online. Association for Computational Linguistics.

Xueqing Liu and Chi Wang. 2021. An empirical study on hyperparameter optimization for fine-tuning pre-trained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2286–2300, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. RoBERTa: A robustly optimized BERT pretraining approach.

Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13480–13488.

Bill MacCartney and Christopher D. Manning. 2008. Modeling semantic containment and exclusion in natural language inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 521–528, Manchester, UK. Coling 2008 Organizing Committee.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Anshuman Mishra, Dhruvesh Patel, Aparna Vijayakumar, Xiang Li, Pavan Kapanipathi, and Kartik Talamadupula. 2020. Reading comprehension as natural language inference:a semantic analysis. In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 12–19, Barcelona, Spain (Online). Association for Computational Linguistics.

Arindam Mitra, Pratyay Banerjee, Kuntal Kumar Pal, Swaroop Mishra, and Chitta Baral. 2020. How additional knowledge can improve natural language commonsense question answering?

Sean Papay, Roman Klinger, and Sebastian Padó. 2020. Dissecting span identification tasks with performance prediction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4881–4895, Online. Association for Computational Linguistics.

Charles Sanders Peirce. 1931. *Collected papers of Charles Sanders Peirce*. Harvard University Press.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC² Workshop*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867. Curran Associates, Inc.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*,

pages 2158–2170, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Mengzhou Xia, Antonios Anastasopoulos, Ruochen Xu, Yiming Yang, and Graham Neubig. 2020. Predicting performance for natural language processing tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8625–8646, Online. Association for Computational Linguistics.

Zihuiwen Ye, Pengfei Liu, Jinlan Fu, and Graham Neubig. 2021. Towards more fine-grained and reliable NLP performance prediction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3703–3714, Online. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021. DocNLI: A large-scale dataset for document-level natural language inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4913–4922, Online. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33.

Yunchang Zhu, Liang Pang, Yanyan Lan, and Xueqi Cheng. 2020. L2r²: Leveraging ranking for abductive reasoning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20.

A Model Hyperparameters

All models were trained for 3 epochs with a weight decay of 0.01. The batch size and learning rate used for each model can be seen in Table 2

Model	Learning rate	Batch size
albert-base-v2	10^{-5}	60
albert-large-v2	10^{-5}	20
bert-base-uncased	$5 \cdot 10^{-5}$	32
bert-large-uncased	10^{-5}	16
distilbert-base-unc.	$9 \cdot 10^{-5}$	128
squeezebert/squeezebert-unc.	$7 \cdot 10^{-5}$	64
google/mobilebert-unc.	$7 \cdot 10^{-5}$	100
google/canine-s	10^{-5}	24
google/electra-small-discr.	$7 \cdot 10^{-5}$	128
google/electra-base-discr.	$3 \cdot 10^{-5}$	64
google/electra-large-discr.	10^{-5}	16
microsoft/mpnet-base	$3 \cdot 10^{-5}$	64
roberta-base	$3 \cdot 10^{-5}$	60
roberta-large	10^{-5}	12
google/bigbird-roberta-base	10^{-5}	40
kssteven/ibert-roberta-base	$3 \cdot 10^{-5}$	64
distilroberta-base	$5 \cdot 10^{-5}$	80

Table 2: The learning rate and batch size that resulted in the best classification accuracy for each model.