# Binary Classification vs. Ranking in Abductive Reasoning

**Esra Dönmez**
University of Stuttgart
st172173@stud.uni-stuttgart.de

**Nianheng Wu**
University of Stuttgart
st176149@stud.uni-stuttgart.de

## Abstract

Abductive reasoning is inference to the most plausible explanation (Bhagavatula et al., 2019). The abductive natural language inference task ($\alpha$NLI) is proposed to assess the abductive reasoning capability of machine learning systems. In $\alpha$NLI dataset, the examples consist of a pair of observations and possible hypotheses. The task is to pick the hypothesis that is the most plausible explanation for a given observation pair. The majority of the approaches explored the task as a classical binary classification, where cross-entropy log-likelihood is used as the loss function. However, the nature of the task suggests that ranking might be a better approach. None of the hypotheses in the dataset are wrong, and they can be seen as more or less probable depending on what they are compared with. In this paper, we fine-tune pre-trained RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2020) models on $\alpha$NLI following both binary classification and ranking approaches [1]. Our results show that when all the variables are kept the same, learning to rank suppresses the binary classification approach across several pretrained models in $\alpha$NLI task. Moreover, following both approaches, we obtain better performance when we use DeBERTa model to learn the contextual embeddings.

## 1 Introduction

In the sense, in which it is used most frequently in the modern literature, abduction refers to the place of explanatory reasoning in justifying hypotheses (Douven, 2021). In this sense, abduction is often thought of as inference to the most plausible explanation. Unlike other types of inference, abduction is considered to be the only logical operation that can introduce new ideas (Peirce, 1965). On that account, we consider abductive reasoning to be an important task, from which AI systems might benefit immensely. Despite the central role abduction plays in natural language understanding (NLU), the task of abductive reasoning has not gained much attention from the AI community until recently. In 2020, Bhagavatula et al. introduced a new task ($\alpha$NLI) encouraging the community to explore abductive reasoning deeper. In $\alpha$NLI, the examples consist of a context and a pair of candidate hypotheses. The context is defined as a pair of observations, $O_1$ and $O_2$, start and end observations respectively. The task is to pick the hypothesis that is the most plausible explanation for a given observation pair. The relation between the hypothesis and the observations in $\alpha$NLI task is defined as

$$\begin{aligned} h^* &= \underset{h^i}{\operatorname{argmax}} P(h^i|O_1, O_2) \\ &= \underset{h^i}{\operatorname{argmax}} P(O_2|O_1, h^i) P(h^i|O_1), \end{aligned} \tag{1}$$

$O_1$ : Jasper told his parents that he wanted a dog.
$O_2$ : His parents decided not to give him a dog.
$H_1$ : **[Most Likely]** Jasper asked his parents, but they were allergic to dogs.
$H_2$ : **[Likely]** Jasper got in trouble the next day.
$H_3$ : **[Unlikely]** Jasper asked his parents, but they were allergic to rabbits.
$H_4$ : **[Least Likely]** He promised he would take care of it.

Figure 1: An example in $\alpha$NLI task. For a given observation pair, there are several hypotheses whose plausibility ranges from most likely to least likely.

[1] Code on: https://github.com/esradonmez/CL-teamlab

where $h^i$ is depended on $O_1$, and $O_2$ is depended on both $O_1$ and $H^i$.

Since the task proposal, several models have been successfully developed for $\alpha$NLI task. The majority of these models approach the task from a classification point of view. In these models, the task is formulated as a binary classification problem, where one hypothesis is chosen to be the correct explanation for a given observation pair in each data point. However, this type of task formulation does not exactly fit the description of abductive reasoning. Classifying observations as correct or false suggest that the hypothesis that is less plausible for a given observation pair has 0 probability of occurrence in that narrative, which is not true. With similar motivation, (Zhu et al., 2020) proposed to leverage ranking in abductive reasoning. In their experiments, they used ESIM (Chen et al., 2016), BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) models as their scoring functions and obtained state-of-the-art results at the time when compared to binary classification.

Recently, a new pre-trained language model DeBERTa (He et al., 2020) was proposed to improve the performance of RoBERTa and other transformers-based models by representing the input with two vectors and applying disentangled attention over them. Different from the other transformers-based models, DeBERTa encodes positional information into the input representation and learns this positional information via disentangled attention. Due to the sequential nature of the task, i.e. an example in $\alpha$NLI dataset is a narrative consisting of a sequence of 3 sentences, positional information plays an essential role in determining the most plausible hypothesis. To solve the task of abductive NLI, we adapted the learning to rank for reasoning (L2R$^2$) framework proposed by Zhu et al. (2020) to use DeBERTa as the scoring function. For comparative analysis, we fine-tuned RoBERTa and DeBERTa on $\alpha$NLI and used the models as the scoring function, both in binary classification and following L2R$^2$ approaches. Our results show that learning to rank framework facilitates higher performance than binary classification on $\alpha$NLI task across several pretrained models. Furthermore, we see improvement in the performance in both binary classification and L2R$^2$ using DeBERTa over RoBERTa.

## 2 Related Work

In document retrieval, learning to rank is a task defined as follows. Assume that there is a query and several documents that might be relevant for this query. In document retrieval (i.e., ranking), given this query, we assign scores to each document according to a ranking function and rank the documents in descending order of the scores. The higher a document ranks in the list, the more relevant we consider it to be for a given query.

In our dataset, there are several possible hypotheses for a given observation pair. The way the task was originally proposed suggests picking a correct hypothesis for each example in the dataset. By doing so, we might ultimately end up confusing the model, as a less plausible hypothesis in one example can be the more plausible one in another. Furthermore, in such an approach, we are inevitably ignoring useful information by not considering all the available hypotheses for a given observation pair when training the model, although they are provided in the dataset.

Pairwise approach to learning to rank has been successfully employed in several machine learning tasks over the years. This approach takes document pairs as instances in learning and formalizes the problem of learning to rank as that of classification by assigning relative relevance scores to these two documents (Cao et al., 2007). Listwise approach to learning to rank, where documents in a list are assigned relative scores and ranked in descending order given a query, was initially proposed by Cao et al. (2007). In 2020, Zhu et al. adapted both pairwise and listwise approaches for $\alpha$NLI task and obtained state-of-the-art performance at the time.

As suggested by Cao et al. (2007), the task of $\alpha$NLI, as it is originally formulated, can be thought of as an incomplete pairwise ranking problem, since only a small part of the possible order of hypotheses are taken into account for classification. As mentioned earlier, this pairwise approach ignores useful information in the dataset. Therefore, we can reformulate the examples into a ranking list, as we have the complete list of possible hypotheses in the dataset, and apply both pairwise and listwise approaches to learning to rank in $\alpha$NLI task. In this paper, we experiment with learning to rank for reasoning (L2R$^2$) approach, which is an adaptation of learning to rank framework for reasoning proposed by Zhu et al. (2020).

## 3 Methodology

### 3.1 Binary Classification

In the binary classification approach, we concatenate the full sequence, which is frequently referred to as a narrative story, to obtain the input to the tokenizer as **concat**($O_1$, $H_1$, $O_2$) and **concat**($O_1$, $H_2$, $O_2$). As labels, we assign 1 to the correct sequence and 0 to the incorrect one.

The encoded input is then fed into the pretrained language model to obtain a contextual embedding for each word in the input, followed by a single linear layer with a cross-entropy loss function (log-likelihood) and a softmax layer for the final prediction.

### 3.2 Learning to Rank for Reasoning (L2R$^2$)

Following the typical structure of learning to rank approach, L2R$^2$ consists of a scoring function to assign a real value score to the hypotheses and a loss function to assess the predictions against the ground-truth labels.

In scoring functions using pretrained language models, such as RoBERTa and DeBERTa, the observations and the hypothesis are concatenated into a full narrative story with a delimiter and a sentinel token. The sequence is then fed into the pretrained language model to obtain a contextual embedding for each token in the input story. To have a vector representation of the input story, mean pooling is applied to the output of the language model. This vector is then fed into a dense layer to obtain a real value score for each input data point.

Just like in the classification setting, the loss function is used here to assess the correctness of the predictions. However, instead of calculating the log-likelihood of the prediction given the ground-truth label, we are comparing the score computed by the final dense layer with the ground-truth score, which is calculated as:

$$\frac{\#(H^i \; occurs \; as \; more \; plausible)}{\#(H^i \; occurs \; for \; this \; observation \; pair)}$$

.

The scoring functions are optimized by minimizing the empirical risk:

$$R(f) = \frac{1}{m}\sum_{i=1}^{m} L(\mathbf{y}^{(i)}, \mathbf{s}^{(i)}), \tag{2}$$

where $L(\mathbf{y}^{(i)}, \mathbf{s}^{(i)})$ is defined differently in each function to evaluate the prediction score for a single query.

Zhu et al. (2020) published their experiments with several loss functions both in pairwise and in listwise approaches. In this paper we experimented with three of their best-performing lost functions: Kullback–Leibler divergence (KLD) (Cao et al., 2007), hinge loss (Herbrich et al., 2000) and ListMLE (Xia et al., 2008).

KLD loss (a listwise loss function) follows the form:

$$L^l(\mathbf{y}, s) = \sum_{j=1}^{N} \frac{e^{y_j}}{\sum_{k=1}^{N} e^{y_k}} log \frac{e^{s_j}}{\sum_{k=1}^{N} e^{s_k}}. \tag{3}$$

ListMLE loss (a listwise loss function) has the form:

$$L^l(\mathbf{y}, s) = -logP(\pi_{\mathbf{y}}, s), \tag{4}$$

where $\pi_{\mathbf{y}}$ is the ground truth permutation.

Lastly, hinge loss (a pairwise loss function) is defined as:

$$L^P(\mathbf{y}, s) = \sum_{y_j > y_k} \phi(s_j - s_k), \tag{5}$$

where $\phi(z) = max\{0, 1 - z\}$ and $y_j > y_k$ means that $H^j$ ranks higher than $H^k$ with regard to the query ($O_1$,$O_2$).

During inference, we choose the hypothesis with the highest score, which satisfies the task of picking the most plausible hypothesis.

## 4 Experiments

In this section, we present our experiments with baseline model, binary classification and following L2R$^2$ approach.

### 4.1 Baseline

To get to know the task, we first implemented a BoW (Bag-of-Words) model using a multilayer perceptron. In our baseline approach, we treat the task as a binary classification problem. We experimented with 2 different types of inputs, i.e. hypothesis-only and fully-connected. In the experiment with hypothesis-only input, we simply extract the hypotheses in the data with their corresponding labels as our input to the model. In our experiment with fully-connected input, we first concatenate the sequence in a given example as **concat**($O_1$, $H_1$, $O_2$) and **concat**($O_1$, $H_2$, $O_2$). As labels, we assign 1 to the correct sequence and 0 to the incorrect one. We then initialize each word in the sequence using GloVe embeddings (Pennington et al., 2014) and apply mean pooling to obtain our input to the model.

Once we obtain our input representations, we pass the input through two dense layers (150 neurons each) followed by a ReLU activation function, and a final softmax layer for prediction. We set the learning rate to 0.01 and a maximum number of epochs to 10 with early stopping. Both experiments (hypothesis-only and fully-connected) converged after the 3rd epoch. On the test set, we obtained 50.6% accuracy with the hypothesis-only and 51.4% accuracy with the fully-connected input.

### 4.2 Binary Classification

Following the original task proposal, we trained a binary classifier to predict the more plausible hypothesis out of two, as they are given in the dataset. In this simple classifier, the input is encoded by a tokenizer designed for the pretrained language model, then fed into the language model (RoBERTa or DeBERTa) to learn the contextual embeddings. The embeddings are then fed into a dense layer followed by a softmax output layer to get the final predictions.

In the RoBERTa classifier, we used RoBERTa-large with the hyperparameters as follows: learning rate = {2e-6, 1e-6}, batch size = 8, warm-up steps = 150 and max. number of epochs = 4. In the DeBERTa classifier, we used DeBERTa-large while keeping all the hyperparameters the same as in the RoBERTa classifier.

### 4.3 L2R$^2$

In this approach, we followed the implementation from Zhu et al. (2020) and built an additional model class (DeBERTa) on top of their work.

In all the experiments with both models, we set our hyperparameters as follows: linear dropout rate = 0.6, max. number of epochs = 10, learning rate = {5e-7, 1e-7}, and training batch size = 1. In all our experiments, we used RoBERTa-large and DeBERTa-large as our scoring functions.

## 5 Results

The results of the models following both approaches are shown table 1.

With RoBERTa-large, we were able to replicate the performance reported by Zhu et al. (2020), i.e., 86.81%, with a slight improvement using KLD. Here, it must be noted that on the test set, they only reported the results of the experiment using the loss function that scored the best on the validation set, i.e., RoBERTa-large + KLD.

| Model | Accuracy |
|---|---|
| RoBERTa-binary | 83.57 |
| **DeBERTa-binary** | **85.78** |
| RoBERTa + KLD | 86.99 |
| RoBERTa + Hinge | 86.24 |
| RoBERTa + ListMLE | 86.17 |
| DeBERTa + KLD | 87.05 |
| **DeBERTa + Hinge** | **88.10** |
| DeBERTa + ListMLE | 87.90 |

Table 1: Results of the models on the test data

Our implementation using DeBERTa-large did not only outperform their results with RoBERTa-large on the test set using KLD, but it also showed improvement when trained with Hinge Loss over KLD.

## 6 Discussion and Future Work

In this paper, we compared binary classification and ranking in $\alpha$NLI task. In $\alpha$NLI task, all the hypotheses are likely to occur, although their plausibility might change depending on the pairs. Therefore, we believe that the nature of the task might fit better into learning to rank framework than binary classification. Following this intuition, we trained two different classifiers using the same pretrained language models to score the choices. In our experiments, we consistently achieved better performance using ranking over binary classification. Furthermore, Zhu et al. (2020) achieved their best performance using RoBERTa in their suggested method (L2R$^2$). Building on their research, we show that DeBERTa facilitates some improvement in L2R$^2$. We hypothesize that representing the input as two vectors (one for content and one for position) and applying disentangled attention over them, as it is implemented in DeBERTa, is better suited for multi-sentence natural language input.

Although we showed that DeBERTa improved the performance in both approaches, we have not yet analyzed if our hypothesis regarding the model architecture is correct. For future work, we suggest comparing the models in other tasks to eliminate the effects of other variables that come with the complexity of the $\alpha$NLI task. Future work could focus on the performance change in different tasks using the contextual embeddings built by different language models.

## References


Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Wen-tau Yih, and Yejin Choi. 2019. Abductive commonsense reasoning. *arXiv preprint arXiv:1908.05739*.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136.

Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2016. Enhanced lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.

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*.

Igor Douven. 2021. Abduction. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2021 edition.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Ralf Herbrich, Thore Graepel, Klaus Obermayer, et al. 2000. Large margin rank boundaries for ordinal regression. *Advances in large margin classifiers*, 88(2):115–132.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Charles Sanders Peirce. 1965. Principles of philosophy and elements of logic. vol. 1. *Collected Papers*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, pages 1192–1199.

Yunchang Zhu, Liang Pang, Yanyan Lan, and Xueqi Cheng. 2020. L2r$^2$: Leveraging ranking for abductive reasoning. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1961–1964.

# Contributions

Esra Dönmez Nianheng Wu

July 29, 2021

## 1 Code:

### Baseline:

**Esra:** Dataset class, evaluation metric, preprocessing file, file reader, training feature creation from glove embeddings
**Nianheng:** Multilayer perceptron, model training function, model evaluation function

### Binary classifier:

**Esra:** Dataset class, training and testing RoBERTa
**Nianheng:** Training and testing DeBERTa

### L2R$^2$:

**Esra:** Training and testing RoBERTa
**Nianheng:** DeBERTa model class, training and testing DeBERTa

## 2 Paper:

### Initial draft:

**Esra:** Abstract, Introduction, Related Work, Discussion and Future Work
**Nianheng:** Methodology, Experiments, Results, References

After the initial draft, we have done several revisions and made several edits to the paper, changing the content and the style almost completely. Therefore, we would like the paper to be considered as a team contribution.