Which natural language inference problems are hard for neural models?

Marc Zoon
5948002

Supervisor:
Lasha Abzianidze

Second reader:
Thijs van Ommen

Bachelor Thesis: 8ECTS

Universiteit Utrecht

Bachelor Artificial Intelligence
Faculty of Humanities
Utrecht University
31^{st} of May, 2021

Abstract

Last decade the interest in natural language inference has increased because it serves as a task to test AI models on natural language understanding. This resulted in several models with new state-of-the-art performance. While overall accuracy on different benchmarks has been increasing steadily, little research is done on specific problem types that are hard to solve. This paper explores different characteristics of the inference problems, resulting in problem types that are hard to solve for models based on certain architectures or trained on specific data set.

Contents

1 Introduction **3**

2 Method **3**
2.1 Model selection 3
2.2 Feature selection 3
2.3 Model evaluation 4

3 Model selection **4**
3.1 Corpora 4
3.2 Architectures 6
3.3 Relative performance 7

4 Features **8**

5 Experiments and results **11**
5.1 Pre-processing 11
5.2 Model performance 11

6 Discussion **14**

7 Future work **15**

1 Introduction

The Natural Language Inference, or NLI for short, task has gained a lot of attention since the introduction of the Stanford Natural Language Inference (SNLI) data set [Bowman et al., 2015]. The goal of this task is to detect if a hypothesis sentence h is neutral to, entailed, or contradicted from a premise p, a task that requires natural language understanding. One example for this task is, given the premise *Four adults eat while sitting on a tile floor*, if the hypothesis *A group of people eat food* is an entailment, neutral, or contradiction, and in this case is an entailment. While researchers come up with new models to compete over the highest accuracy scores and focus on the quantitative measures, qualitative measures are often left behind. Thus little is known about why a model does or doesn't solve certain problems. Qualitative analysis has been done before on single models [Pavlick and Kwiatkowski, 2019]. However, to get a good understanding of common problems, an analysis over a wide range of models, varying in architectures and training corpora, should be done. A way forward is to look at what types of problems different, widely-used models fail on, and compare their different points of failure. This would give an insight into where the problem lies: the training data, the model architecture, or possibly both. This thesis will explore a few types of problems that may cause trouble for NLI models. All annotated data collected during the research is made publicly available on GitHub[1].

2 Method

The process of finding hard natural language inference problems for neural models can be divided into two parts: NLI model selection, and characteristic feature selection for NLI problems. This section will briefly describe on these parts. Sections 3 and 4 will provide a more detailed description.

2.1 Model selection

NLI models will be selected based on various criteria. Since this is an experiment that attempts to find problems that are hard in general for neural models, the selected models have to vary in architecture and training data. Then, to verify that the models actually behave differently from each other, each pair of models will be assigned a similarity score (the percentage of sentence pair where both models predict the same label).

The selected models are then evaluated on the test set of the SNLI corpus. These predictions, combined with the features, are later used to determine if problem types are hard or not.

2.2 Feature selection

In order to categorize sentence pairs, a number of features are selected. These features represent problem types that intuitively could influence the performance of a model (e.g. the occurrence of a negation). Then, using Python and

[1] `https://github.com/MarcZoon/hardNLIproblems`

the spaCy library, each sentence pair is assigned a true/false value for each of the feature categories.

2.3 Model evaluation

In order to classify problems as hard, the NLI models' predictions are analyzed

3 Model selection

We select the NLI models that are based on widely-used pre-trained language models. Most of these models are based on transformers [Wolf et al., 2020], XLNet and ELECTRA being the two exceptions. To say something useful about the relative performance on different criteria the models have to differ to some extent from each other. To accomplish this models have to be selected based on their underlying architectures and the used corpora. Due to the popularity of BERT-like models, a large part of well-performing models are BERT-like, and is visible in the selection of models. To allow for easy reproduction of the results, the selection of models will be limited to pre-trained models available on Hugging Face. Table 1 lists the different combinations of corpora and architectures used in this analysis. All models will be referred to by the name of the architecture used for the rest of the thesis, with the exception of the two RoBERTa models, which will have an additional tag, (1) or (2), to distinguish between them.

All of the used pre-trained models have accuracies similar to what can be expected from their papers. Table 1 lists a full overview of the accuracy scores.

Architecture	SNLI	MNLI	ANLI	Accuracy
BERT base	*			90.4
RoBERTa large (1)		*		88.3
RoBERTa large (2)	*	*	*	91.8
DistilRoBERTa base	*	*		90.0
DeBERTa xlarge		*		89.7
ALBERT xxlarge	*	*	*	91.9
BART large	*	*	*	92.0
ELECTRA large	*	*	*	91.1
XLNet large cased	*	*	*	91.6

Table 1: Overview of models used in this experiment and their accuracy scores on the SNLI test set. Each model is represented in a row, where a * in a column means the model is trained on that data set.

Sections 2.1 and 2.2 will discuss the used corpora and architectures. Section 2.3 will briefly touch on how the different models perform with respect to each other.

3.1 Corpora

SNLI [Bowman et al., 2015]

The *Stanford Natural Language Inference* data set was introduced by Bowman et al. as a new data set to be used for natural language inference. With its

570K sentence pairs, it was, at the time, the largest publicly available data set. Besides the increased number of sentences, it also attempts to solve a different issue. Earlier corpora had issues with determining the correct semantic label. Some cases where a case could be made for both a neutral or contradiction label (e.g. *A boat sank in the Pacific Ocean - A boat sank in the Atlantic Ocean*), resulted in inconsistent methods of labeling sentence pairs. To counter this problem they set a constraint on how sentences were to be written; each sentence pair has to be written from the same perspective.

For the creation of the sentence pairs workers from Amazon Mechanical Turk were presented a premise, a sentence from the Flick330k corpus [Young et al., 2014], and were tasked to write a hypothesis for each of the labels. To validate the created sentence pair, each was then labeled by four other annotators, and the most common label was set as the correct golden label.

MNLI [Williams et al., 2018]

The *Multi-Genre Natural Language Inference* data set is a more recent data set following the structure of SNLI. While it 'only' contains 433k sentence pairs, thus smaller than the SNLI data set, the new data set is better suited for natural language inference due to some other improvements. First of all, the sentences are categorized by 10 different genres. This makes it easier to evaluate a model's performance on specific topics. The included sentence pairs also have more variation in difficulty. This allows training and evaluating models on sentences that cover all complexity levels of the language. After an evaluation comparing MNLI and SNLI, the MNLI data set represents a more difficult task.

The data collection for MNLI is done in a similar fashion to SNLI. This time, instead of using Amazon Mechanical Turk, Hybrid was used. The premises were taken from existing text sources, and a human annotator was asked to create three hypotheses, again, one for each label. However, this time the annotator was also presented with example premises and hypotheses specific to each genre.

Adversarial NLI [Nie et al., 2019]

Unlike SNLI and MNLI, *Adversarial Natural Language Inference* is not a static data set. It is a, possibly forever, evolving data set that tries to find more sentence pairs that earlier models failed on. The general idea introduced is to present a human annotator with a premise, and task them to write a hypothesis that would be wrongly labeled by a pre-trained model, and the reason why the annotator the model has wrongly labeled the hypothesis. These examples are then validated to be sure the model is mistaken, and not the annotator.

The method used to create the data set introduced in the paper consists of three rounds.

To start, a BERT model was trained on the SNLI and MNLI corpora. Annotators are then provided with multi-sentence premises from the HotpotQA data set [Yang et al., 2018], and are given the task described above.

In the second round, instead of BERT, RoBERTa was trained on SNLI, MNLI, FEVER [Thorne et al., 2018], and the data collected in the first round. To prevent annotators from exploiting vulnerabilities of a single model, a set of models was trained based on random seeds. Again, a set of contexts, or

premises, was selected from the HotpotQA data set and provided to the annotators through the earlier described procedure.

The third round follows the same procedure as the second, with an expanded data set. They added texts from news (extracted from Common Crawl), fiction [Mostafazadeh et al., 2016][Hill et al., 2015], spoken formal text [Ide et al., 2010], and causal or procedural text from WikiHow.

These three rounds resulted in a data set of 103k sentences, with more saved for later rounds. This is smaller than both SNLI and MNLI, but, due to the nature of the included sentences, will be more useful in natural language inference tasks.

3.2 Architectures

BERT [Devlin et al., 2019]

BERT, or *Bidirectional Encoder Representations from Transformers*, uses the transformers architecture introduced by [Vaswani et al., 2017]. BERT pre trains deep bidirectional representations from unlabeled text, and as a result, can supply state-of-the-art models on various tasks by adding just one additional output layer. Thus, BERT itself is not meant to be used for any natural language inference tasks.

BERT is almost identical to the original architecture by Vaswani et al, and thus, follows the encoder-decoder structure.

RoBERTa [Liu et al., 2019]

RoBERTa, short for *Robustly optimized BERT approach*, is the result of a replication study on BERT. They claim that, despite BERT being a great performer, it was undertrained. By evaluating hyperparameters and training set size, RoBERTa outperforms all models released after BERT.

DistilRoBERTa [DistilRoBERTa]

DistilRoBERTa is a *distilled* version of the *RoBERTa* base model. It follows the training procedure introduced for DistilBERT [Sanh et al., 2019]. Sanh shows that the distilled BERT model reduces the size of BERT by 40%, is 60 % faster, and retains 97% of its performance. They accomplish this by leveraging knowledge distillation during the pretraining phase. While larger models, like BERT, are able to learn inductive biases during pretraining, distilled models miss out on those advantages. To compensate for these losses, DistilBERT introduces three new methods; triple loss combining language modeling, distillation, and cosine-distance losses.

The used distilled version of RoBERTa is twice as fast as the original RoBERTa model.

DeBERTa [He et al., 2020]

To improve over BERT, DeBERTa, or *Decoding-enhanced BERT with disentangled attention*, uses two new techniques. First is the disentangled attention mechanism. Here, each token in a sequence is represented by two vectors, representing the token contents, and its position relative to another token.

ALBert [Lan et al., 2019]

ALBert is *A Lite Bert* model that does not necessarily attempt to increase performance in terms of accuracy but in terms of speed. They introduce two methods of parameter reduction to reduce memory usage and increase the training speed. The first is a factorized embedding parameterization. They decompose the single vocabulary embedding matrix into two smaller matrices, separating the size of the hidden layers from the size of the vocabulary embedding. Thus, they allow easier growth of the hidden layers, without significantly affecting the size of the vocabulary embedding. Then they use a cross-layer parameter sharing technique, preventing the number of parameters from growing with the depth of the network. These techniques allow for a reduction in the number of parameters without significantly reducing the model performance

BART [Lewis et al., 2019]

Just like BERT, BART uses the standard sequence-to-sequence Transformers architecture from [Vaswani et al., 2017] and thus have very similar architectures. While very similar, there still are two differences. First, in each layer the decoder performs cross-attention over the final hidden layer of the encoder, this does not happen in BERT. But, BERT has a feature not included in BART, namely an additional feed-forward network before word prediction. In the end, BART has a roughly 10% increase in parameters compared to an equivalent BERT model.

ELECTRA [Clark et al., 2020]

ELECTRA improves over other pre-trained models in how it handles token replacements. In training, models like BERT the input is corrupted and replaced by [MASK], and the model is then tasked to reproduce the original tokens. This produces good results, but at the cost of requiring a large amount of computing power. ELECTRA uses a method more sample-efficient method called replaced token detection. Here the tokens are replaced by plausible alternative tokens, and then the model has to predict whether or not a token has been replaced. This makes better use of samples, since it trains on every token, and not just the those that are masked.

XLNet [Yang et al., 2019]

While models like BERT often perform better than those that rely on auto-regressive language modeling, due to the reliance on corrupting the input the dependencies between the masked tokens are lost and suffer from a pre train-finetune discrepancy. With XLNet, a new generalized auto-regressive pretraining method was introduced that overcomes the limitations of BERT-like models.

3.3 Relative performance

Since most of these models are similar to, or expansions of BERT, and are trained on combinations of the SNLI, MNLI, and ANLI corpora, the similarity between the models has been analyzed. The similarity score is the percentage of predictions where the two compared models predict the same label. With accuracy scores of 90%, the lower bound for these scores is 0.8. The scores for

	BERT	RoBERTa (1)	RoBERTa (2)	DistilRoBERTa	DeBERTa	ALBERT	BART	XLNet	ELECTRA
BERT	100	88	**93**	**93**	89	92	**93**	**93**	91
RoBERTa (1)	88	100	91	89	**92**	90	90	90	89
RoBERTa (2)	93	91	100	93	92	**95**	**95**	**95**	93
DistilRoBERTa	**93**	89	**93**	100	89	92	**93**	**93**	91
DeBERTa	89	**92**	**92**	89	100	91	91	92	90
ALBERT	92	90	**95**	92	91	100	**95**	94	93
BART	93	90	**95**	93	91	**95**	100	**95**	93
XLNet	93	90	**95**	93	92	94	**95**	100	93
ELECTRA	91	89	**93**	91	91	**93**	**93**	**93**	100

Table 2: Similarity scores between the analyzed models as the percentage of equally labeled sentence pairs. Per row: bold faced numbers are the most similar models, red numbers are least similar.

these models are in the range from 0.88 to 0.95, and thus, being only 0.08 (at minimum) above the lower bound, the selected models differ enough from each other. A full overview of the relative accuracy scores is available in table 2.

4 Features

Due to the nature of the task, where the occurrence of a certain feature in the premise can be vastly different than when that same feature would occur in the hypothesis, features have to be specified such that they are directional. The following sections will, for each feature, discuss what they are, and why they may be hard to classify for natural language inference models. An overview of the features, their occurrence, and the number of sentence pairs is available in table 3.

Negations

A sentence contains a negations if it contains the word *not*, including other variations such as *don't* and *can't*. While there are other words that could be considered a negation, like no or none, they are also used in other contexts (e.g. *There are no cars on the parking lot*) and therefore are not included. Since a sentence pair with a negation often is a contradiction, it makes sense that a model could correlate the two, causing a mislabeling of entailed and neutral sentences. One thing to keep in mind is that there are only 102 unique sentences in the SNLI test set that contain a negation, and therefore also very few (1%) sentence pairs where there is at least one occurrence of a negation.

Feature	Group	%	Count
Negation	No-No	98.8	9706
	No-Yes	0.9	91
	Yes-No	0.2	25
	Yes-Yes	<0.1	2
	¬(No-No)	1.2	118
Sentence	Yes-Yes	68.0	6684
	Yes-No	7.6	744
	No-Yes	20.5	2018
	No-No	3.8	378
	¬(Yes-Yes)	32.0	3140
Voice	Act-Act	97.3	9554
	Act-Pas	1.1	115
	Pas-Act	1.4	142
	Pas-Pas	0.1	13
	¬(Act-Act)	2.7	270
Quantifiers	No-No	89.2	8786
	No-Yes	3.9	379
	Yes-No	5.9	581
	Yes-Yes	0.8	78
	¬(No-No)	10.6	1038
Length (words)	Equal	5.6	550
	Premise	83.8	8236
	Hypothesis	10.6	1038
Length (noun chunks)	Equal	20.0	1952
	Premise	73.8	7252
	Hypothesis	6.3	620

Table 3: Distribution and occurrences of sentence pairs per feature.

Complete sentence

Whether a sentence is complete or not can be determined by the root of the parse tree, available in the SNLI data set, of a sentence; if the root is equal to *S*, the sentence is a complete sentence. Incomplete sentences may miss crucial elements, such as the subject or verb, in a way that the sentence still makes sense to a human reader but is harder for a model to understand. This lack of information can make natural language inference tasks harder. In the SNLI test set, there are 11046 complete and 1915 incomplete sentences, and 68% of the sentence pairs contain two complete sentences.

Sentence voice

Since active sentences are more common than passive sentences in natural language, the sentence voice may be another important feature to look at. Not necessarily because passive sentences are harder to understand, but since they are less common than active ones, corpora simply include less of them. This is

clearly visible in the SNLI data set, only 180, or 1%, are passive sentences.

Passive sentences can easily be used by annotators to paraphrase the premise to create an entailment. If this is the case, models could correlate the active voice of the premise and the passive voice of the hypothesis with an entailment, making it prone to errors when the hypothesis is either neutral or a contradiction.

Quantifiers

The quantifiers feature includes sentences that contain quantifiers, words like *some*, *none*, *all*. The occurrence of a quantifier in a sentence can be an advantage in natural language inference tasks. These words can easily be used to distinguish between sentences such as *some children are playing* and *all children are playing*. Cases like these make it easy to see why features have to be defined directionally. When some children play, it is not necessarily true that all of them play. But when all of the children play, then we can be sure that some of them are playing.

On the other hand, quantifiers single words that have a lot of semantic value. In the sentence *The men are both wearing glasses* the word *both* adds a lot of value. It implies that there are exactly two men in the picture, and that the two of them are wearing glasses. There could be other people in the picture, but they could not be men, and whether or not they wear glasses also would not matter. All of this information is added to that short sentence by the word *both*, and could be hard for natural language inference models to label as entailment, neutral, or contradiction.

Sentence length (words)

While longer sentences may be harder to parse, and thus harder to assign a label to, the relative sentence length might be a better indication. Since longer sentences contain more information than shorter ones, it makes sense that longer sentences are less likely to not entail from shorter ones. A longer hypothesis entailing from the premise is still possible, and does occur in the SNLI data set. Some are sentences a person could reasonably use; *A biker races* as the premise, *A person is riding a bike* as the hypothesis. But others are vague descriptions using an excessive number of words to describe simple events; premise: *A dog jumping for a Frisbee in the snow.*, hypothesis: *An animal is outside in the cold weather, playing with a plastic toy.*) The second example is clearly an entailment, it simply generalizes the premise.

Sentence length (noun chunks)

To somewhat compensate for longer descriptions of the same thing, it is possible to use noun chunks instead of the number of words. When a longer sentence has the same meaning as a shorter sentence (or at least parts of the sentences have the same meaning), the number of noun chunks is a better indication of the amount of information in the sentences. A *cruise ship* can be described as a *large passenger ship* without adding any additional information. This does not solve the problem we find with the first example of the previous section (*A biker races* vs. *A person is riding a bike.*) But should the second sentence

entail from the first one? A person can be a biker, without being on a bike! Thus, the racing biker could be a person that happens to also be a biker but is participating in a foot race. Despite the implications this can have for these sentence length features, it will not be further discussed here, since this is a semantics problem out of the scope of this thesis.

Using the number of noun chunks in a sentence as a measure compared to the number of words has moved some sentences to a different group. The share of sentence pairs of equal length has increased from 5.6% to 20.0%, with a decrease of 10.0% for those where the premise is longer, and a decrease of 4.6% of those sentences where the hypothesis is longer.

5 Experiments and results

The following sections will discuss the pre-processing done for the features, and model performance on the different problem types. The models are run on the SNLI test set ignoring sentence pairs where there is no gold label, those sentence pairs where the gold label is set as '-', then 9824 out of the 10k sentence pairs remain. Only those results that stand out and have some significance will be discussed, but a complete overview of the results is available in table 4.

5.1 Pre-processing

All of the linguistic features in this thesis are extracted from the SNLI data set using basic Python (v3.8) and the spaCy (v3.0) library using the *en_core_web_trf* pipeline. The Negations, sentence voice, and quantifiers are found using spaCy matchers. The number of noun chunks in a sentence corresponds to the *noun_chunks* property in spaCy. Sentence length in words corresponds to the number of tokens in a sentence, ignoring those where the part of speech tag corresponds to punctuation. Whether a sentence is complete is read *sentence_parse* field in the SNLI data set. These features are acquired in a single iteration over all the sentence pairs.

5.2 Model performance

Negations

The SNLI data set includes very few negations, therefore these results possibly do not translate well to other corpora where negations are more common. While it is interesting to see that all models correctly labeled all sentence pairs where both sentences have a negation, there are only two of these sentence pairs, and therefore will not be discussed due to a lack of data.

Despite also having very few entries, the sentences where there is an occurrence of a negation in one of the sentences are worth mentioning. It is obvious how the transformer-based models (all but XLNet and ELECTRA) really suffer from the lack of sufficient data. They lose, on average, 8.5% in accuracy in pairs where the premise contains a negation, while the more data-efficient XLNet and ELECTRA do not lose at all, or even increase accuracy by 1%. On pairs where only the hypothesis contains a negation, all models perform similarly, with a 4% loss compared to sentence pairs where neither sentence contains a negation.

Feature	Group	Distribution	All models (avg)	BERT^{S}	RoBERTa (1)^{M}	RoBERTa (2)^{SMA}	DistilRoBERTa^{SM}	DeBERTa^{M}	ALBERT^{SMA}	BART^{SMA}	XLNet^{SMA}	ELECTRA^{SMA}
Overall			90.8	90.4	88.3	91.8	90.0	89.7	91.9	92.0	91.6	91.1
Negations	No-No	98.8	0.0	+0.1	0.0	+0.1	0.0	0.0	0.0	+0.1	0.0	0.0
	No-Yes	0.9	-4.1	-3.6	-2.6	-3.9	-2.1	-4.0	-5.0	-6.3	-4.8	-4.3
	Yes-No	0.2	-6.1	-10.4	-8.3	-7.8	-10.0	-1.7	-11.9	-8.0	+0.4	+0.9
	Yes-Yes	<0.1	+9.2	+9.6	+11.7	+8.2	+10.0	+10.3	+8.0	+8.4	+8	+8.9
	¬(No-No)	1.2	-4.4	-4.8	-3.6	-4.6	-3.6	-3.3	-6.3	-6.4	-3.5	-3.0
Sentence	Yes-Yes	68.0	0.0	0.0	0.0	+0.1	0.0	-0.2	+0.2	0.0	0.0	0.0
	Yes-No	7.6	-0.4	+0.1	-1.1	+0.5	-0.8	-0.7	-0.8	0.0	0.0	-0.1
	No-Yes	20.5	0.0	+0.1	+0.1	-0.2	+0.2	-0.2	-0.6	+0.2	+0.4	+0.3
	No-No	3.8	-0.3	+1.9	-0.7	0.0	-1.1	0.0	+1.0	-1.5	-0.3	-1.7
	¬(Yes-Yes)	32.0	-0.1	+0.3	-0.3	0.0	-0.2	-0.3	-0.4	0.0	+0.2	0.0
Voice	Act-Act	97.3	0.0	+0.1	0.0	0.0	0.0	+0.1	0.0	0.0	0.0	0.0
	Act-Pas	1.1	+0.3	+1.7	-1.3	+1.2	-0.4	-0.4	+2.0	-1.6	+2.3	-0.7
	Pas-Act	1.4	-2.7	-5.2	+0.4	+0.3	-4.8	-5.2	-3.9	-1.1	-2.9	-1.0
	Pas-Pas	0.1	-2.8	+1.9	**+11.7**	-7.2	-5.4	**+10.3**	-7.3	-15.1	-7.0	-6.4
	¬(Act-Act)	2.7	-1.4	-1.9	+0.2	0.0	-3.0	-2.3	-1.5	-2.0	-0.9	-1.1
Quantifiers	No-No	89.2	+0.2	+0.2	+0.2	+0.2	+0.2	+0.4	-0.2	+0.2	0.0	+0.3
	No-Yes	3.9	-1.9	-1.5	-1.2	-1.3	-2.1	-1.3	-1.4	-2.6	-1.6	-3.5
	Yes-No	5.9	-1.2	-0.2	-2.6	-0.9	-1.9	-3.1	-1.2	0.0	+0.7	-1.3
	Yes-Yes	0.8	-1.2	-4.5	-1.1	**+1.8**	-1.5	-5.1	-3.4	-2.3	+0.7	-1.3
	¬(No-No)	10.6	-1.5	-1.0	-2.0	-0.9	-1.9	-2.6	-1.4	-1.2	-0.2	-2.3
Length (words)	Equal	5.6	-0.3	-0.5	-0.1	-0.3	-0.2	**+1.0**	+0.8	-0.7	+0.6	-0.6
	Premise	83.8	+0.1	+0.3	-0.1	+0.2	0.0	-0.2	0.0	+0.2	+0.2	0.0
	Hypothesis	10.6	-0.8	-0.5	+0.2	-0.7	0.0	+**1.4**	0.0	-0.6	-1.2	-0.3
Length (n. ch.)	Equal	20.0	+0.2	-0.1	+0.7	-0.2	-0.4	**+1.9**	+0.4	0.0	+0.3	0.0
	Premise	73.8	-0.1	+0.2	-0.2	0.0	0.0	-0.4	-0.2	+0.1	+0.1	0.0
	Hypothesis	6.3	-0.4	-1.0	-1.0	+0.6	+0.5	-0.5	+0.5	-0.5	-1.3	+0.2

Table 4: The accuracy scores per feature and feature-class (in % relative to a model's overall score). Boldfaced and red values are those mentioned in the text. In superscript are the data sets the model is trained on, S=SNLI, M=MNLI, A=ANLI

When looking at all sentence pairs where at least one of the sentences contains a negation, all models perform worse, on average 4.4% lower, compared to their overall score. Three models lose more than 4.4% in accuracy, BERT, ALBERT, and BART. This may seem surprising for ALBERT and BART at first since they are trained on SNLI, MNLI, and ANLI. However, when comparing the two models using the RoBERTa large architecture, RoBERTa (1) only loses 3.6% in accuracy compared to RoBERTa (2)'s 4.6%. This difference could indicate that the MNLI data set used to train RoBERTa(1) is better suited to solve these types of problems. But then, why do XLNet and ELECTRA perform relatively well compared to the other models trained on ANLI? If it is the case, which is very likely, that the number of negations in the ANLI data set is very small, then XLNet and ELECTRA have an advantage over the other models. The XLNet and ELECTRA architectures were designed in such a way that they are able to learn from a smaller amount of samples.

Complete sentence

Out of the analyzed models, most are very insensitive to whether or not a sentence is complete. This can have a few causes. First of all, the incomplete sentences still provide enough information for a human to make a correct inference. If models can make correct inferences from the same amount of information as humans, which is true for all sentence pairs in the SNLI data set where the gold label is not '-', then it does not matter if a sentence is complete or not in the context of the SNLI data set.

On sentence pairs where at least one of the sentences is incomplete, there is no model that gains or loses more than 1% in accuracy, and most do not change at all (<1%). BART is the only model that loses 1.5% on any of the feature types, the one where neither of the sentences is complete. In general, models seem to be insensitive to whether or not a sentence is complete.

Sentence voice

As noted earlier, passive sentences are uncommon (around 1.4% of the unique sentences, 3% of sentence pairs with at least one passive sentence). With few samples to train on, it is unsurprising that most models perform worse on these types of problems. What is surprising, however, is that the models which are trained exclusively on the MNLI data set manage to correctly label all the Passive-Passive sentence pairs. Despite the low number of sentence pairs with this feature, it is a massive difference compared to the other models, that, with the exception of BERT, score lower. This improvement on Passive-Passive sentence pairs does not translate to sentence pairs with only one passive sentence, where RoBERTa (1) scores similar to their overall accuracy, and DeBERTa loses 5.2% on Passive-Active sentence pairs.

BART achieves an accuracy 15.1% lower than its overall score on Passive-Passive sentence pairs. While other models trained on the same corpora also perform worse, BART's accuracy drop by at least 8% more, more than double, compared to the others. It seems like the architecture is less suited to deal with these problems.

Looking at the accuracy scores for all problems with at least one passive sentence, all models, with the exception of BERT, score similar or worse compared to their overall accuracy.

While the increase in performance for RoBERTa (1) and DeBERTa have a simple possible explanation, namely that MNLI is a data set better suited for these problems, the high decrease in accuracy for BART has no easy explanation. The problem most likely lies in the BART architecture, despite being very similar to other transformer-based models.

Quantifiers

When introducing the quantifiers feature, two possible outcomes were mentioned. As it turns out, sentence pairs containing quantifiers are harder to correctly label than those that do not. The model that loses the most in accuracy in a single measure is DeBERTa, losing 5.1% on the sentence pairs where both sentences contain a quantifier. Right behind are BERT, ALBERT, and ELECTRA, losing around 4% on the same problems. This is not surprising for BERT, DeBERTa, and ALBERT, since there are very few sentence pairs with

this feature. ELECTRA, however, is designed to be efficient with sample size, and a better accuracy should be expected. Meanwhile, RoBERTa (2), trained on the same corpora as ALBERT and ELECTRA, improves by 2% on these problems.

Looking at the accuracy of all sentence pairs with at least one quantifier, DeBERTa performs worst again, with a 2.6% loss on its overall score. The rest of the models score slightly, but consistently worse when there is a quantifier.

Sentence length (words)

Looking at relative sentence length based on the number of words the accuracy scores do not change by much, <1% for most models. The exceptions are DeBERTa and XLNet. DeBERTa scores 1% higher on sentences of equal length, and 1.4% on sentence pairs with a longer hypothesis. XLNet scores 1.2% lower on sentence pairs where the hypothesis is longer.

Sentence length (noun chunks)

When we use the number of noun chunks instead of the number of words to determine relative sentence length, we can observe a small change in the accuracy scores. Overall, the accuracy scores increase from -0.3% to +0.2% relative to the overall scores. This is not surprising for a few reasons. First of all, the number of sentence pairs with equal length based on noun chunks is four times larger than the number based on the number of words, increasing the number of training samples available. A second reason is that when the amount of information in both sentences is equal, it is easier for a model to compare the information.

This increase in performance for sentences of equal length comes from sentences that were in the other categories when using number of words, thus, on sentence pairs where the premise and hypothesis are not of equal length, the models achieve a lower accuracy.

When looking at individual models, the same observations can be made on almost all models. DeBERTa gains the most from this change, where its accuracy on equal sentences goes from +1.0% to +1.9%, but also loses the most on those pairs where the hypothesis is longer, from +1.4% to -0.5%.

In contrary to the other models, BERT, RoBERTa (1), and XLNet now score lower on sentence pairs with a longer hypothesis.

6 Discussion

The following paragraphs will discuss the linguistic features that were found to have an impact on the accuracy scores, and possible reasons as to why models have trouble with them.

Considering all of the explored types of problems, we observe the biggest loss in accuracy in problems where at least one of the sentences contains a negation. It is hard to say why exactly models fail on these problems, but there are a few possible explanations. Since the number of negations in the corpora is so low, models have very few samples to learn from. A second possibility is than the problem is actually hard for natural language inference.

Other hard sentence pairs to correctly label are those containing passive sentences. It is surprising to see that models that are trained on either SNLI or MNLI perform better than their overall scores on sentence pairs where both sentences are passive. What is surprising at first glance is that DistilRoBERTa, which is trained on both SNLI and MNLI, performs worse. It could indicate that certain kinds of problems, in this case, sentences in the passive voice, accuracy can be highly increased by using a better data set, but they require larger architectures to do so.

The third type of problem that proves a challenge for the models are the sentences with quantifiers. On these problems, there is no recognizable pattern as to how models perform. They score worse across the board, with the exceptions of RoBERTa (2) on sentence pairs where both sides have a quantifier, and XLNet on sentence pairs that contain a quantifier in the premise. Therefore it is not possible to tell where these losses in accuracy originate from.

When we compare two models with the same architecture, RoBERTa (1) (trained on MNLI) and (2) (trained on SNLI, MNLI, and ANLI), we can see the importance of the used training data. Overall RoBERTa (2) performs 3.5% better. However, when we look at sentence pairs where both sentences are in passive voice, suddenly RoBERTa (1) outperforms RoBERTa (2), and also almost all other models, DeBERTa being the only exception. Coincidentally, DeBERTa is also trained on the MNLI data set. This indicates that the MNLI data set is very good at solving this type of problem. This shows that different corpora have their own strengths and weaknesses and that even the corpora with state-of-the-art performance are not able to consistently solve every type of problem.

7 Future work

To get a better understanding of common hard problems for natural language inference models there are multiple directions further research can go. First of all, the SNLI test set used to evaluate the models is limited in size, and therefore the number of samples for certain problem types, specifically the negations, are very uncommon. Using a multitude of corpora (e.g. SNLI, MNLI, and ANLI) would increase the number of sentence pairs to test on, resulting in more reliable results. An even better approach, despite being an enormous task, is to create a new test set designed in such a way that certain linguistic features that may prove to be a challenge occur more often.

Secondly, there is room for improvement in regards to the selected linguistic features. One can test on entirely different features (e.g. antonyms or verb tense), or improve on those explored here. Let's take a look at the quantifiers. Here, the list of selected contains 16 quantifiers, and all are grouped together. It remains unknown on which quantifiers models fail more often. There could be just a few specific ones that are hard for these models, while the rest are easy to correctly label and pull the accuracy scores up on these problems.

And lastly, this experiment has only categorised sentence pairs based on a single feature. While it is a good place to start, it is not unlikely that specific feature combinations are more indicative of a problem's difficulty. While this allows a more specific analysis of the task, it also brings a new challenge. As it is, some single features occur infrequently. Adding additional constraints to

these features would then further reduce the already low number of sentence pairs.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. *arXiv*, Aug 2015. URL https://arxiv.org/abs/1508.05326v1.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. *ArXiv e-prints*, Mar 2020. URL https://arxiv.org/abs/2003.10555v1.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

DistilRoBERTa. distilroberta-base · Hugging Face, Apr 2021. URL https://huggingface.co/distilroberta-base. [Online; accessed 13. Apr. 2021].

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *ArXiv e-prints*, Jun 2020. URL https://arxiv.org/abs/2006.03654v4.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. *ArXiv e-prints*, Nov 2015. URL https://arxiv.org/abs/1511.02301v4.

Nancy Ide, Collin F Baker, Christiane Fellbaum, and Rebecca J Passonneau. The manually annotated sub-corpus: A community resource for and by the people. In *Proceedings of the ACL 2010 conference short papers*, pages 68–73, 2010.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *ArXiv e-prints*, Sep 2019. URL https://arxiv.org/abs/1909.11942v6.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *ArXiv e-prints*, Oct 2019. URL https://arxiv.org/abs/1910.13461v1.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv e-prints*, Jul 2019. URL https://arxiv.org/abs/1907.11692v1.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A Corpus and Evaluation Framework for Deeper Understanding of Commonsense Stories. *ArXiv e-prints*, Apr 2016. URL https://arxiv.org/abs/1604.01696v1.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A New Benchmark for Natural Language Understanding. *arXiv*, Oct 2019. URL https://arxiv.org/abs/1910.14599v2.

Ellie Pavlick and Tom Kwiatkowski. Inherent Disagreements in Human Textual Inferences. *Transactions of the Association for Computational Linguistics*, 7: 677–694, Mar 2019.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv e-prints*, Oct 2019. URL https://arxiv.org/abs/1910.01108v4.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for Fact Extraction and VERification. *ArXiv e-prints*, Mar 2018. URL https://arxiv.org/abs/1803.05355v3.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *ArXiv e-prints*, Jun 2017. URL https://arxiv.org/abs/1706.03762v5.

Adina Williams, Nikita Nangia, and Samuel Bowman. 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, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://www.aclweb.org/anthology/N18-1101.

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 M. Rush. Transformers: State-of-the-Art Natural Language Processing. *ACL Anthology*, pages 38–45, Oct 2020. doi: 10.18653/v1/2020.emnlp-demos.6.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. *ArXiv e-prints*, Sep 2018. URL https://arxiv.org/abs/1809.09600v1.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *ArXiv e-prints*, Jun 2019. URL https://arxiv.org/abs/1906.08237v2.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. doi: 10.1162/tacl_a_00166. URL `https://www.aclweb.org/anthology/Q14-1006`.