How does Punctuation Affect Neural Models in Natural Language Inference

Adam Ek **Jean-Philippe Bernardy** **Stergios Chatzikyriakidis**
Centre for Linguistic Theory and Studies in Probability
Department of Philosophy, Linguistics and Theory of Science
University of Gothenburg
{adam.ek,jean-philippe.bernardy,stergios.chatzikyriakidis}@gu.se

Abstract

Natural Language Inference models have reached almost human-level performance but their generalisation capabilities have not been yet fully characterized. In particular, sensitivity to small changes in the data is a current area of investigation. In this paper, we focus on the effect of punctuation on such models. Our findings can be broadly summarized as follows: (1) irrelevant changes in punctuation are correctly ignored by the recent transformer models (BERT) while older RNN-based models were sensitive to them. (2) All models, both transformers and RNN-based models, are incapable of taking into account small relevant changes in the punctuation.

1 Introduction

In recent years models for Natural Language Inference (NLI) have reached almost human-level performance. These models frame inference as a classification problem, whose input is a premise/hypothesis pair. It has been noted that small changes in the pair, can flip the prediction (Glockner et al., 2018). In this paper, we explore the effect of punctuation[1] in neural models in natural language inference.

Small changes in a premise/hypothesis pair are of two kinds. First, the change can be of an irrelevant kind. For example, we can expect that removing a sentence-final stop should not change the meaning of a (final) sentence. Second, a textually small change could flip the relationship between hypothesis and premise. For example, adding a negation word is a small textual change that has a lot of semantic content. But it is not only words that can have a large impact on the meaning of a sentence. Commas, for example, may indicate which words belong together and which do not in a list. Ideally, an NLI model should be insensitive to changes of the first kind, but still, properly recognize changes of the second kind.

In this paper, we test both hypotheses for the case of punctuation. Namely:

- (H1) Deep-learning based classifiers are sensitive to irrelevant punctuation.
- (H2) Deep-learning classifiers take relevant punctuation into account correctly.

This work is part of the larger question concerning the ability of NLI models to generalize. There are a number of papers that report several problems of generalizability: Glockner et al. (2018) have shown that several NLI models break considerably easily when, instead of tested on the original SNLI (Bowman et al., 2015) test set, they are tested on a test set which is constructed by taking premises from the training set and creating several hypotheses from them by changing at most one word within the premise. Talman and Chatzikyriakidis (2018) show that NLI models break down when one trains in one dataset, but then test on the test set of a similar dataset (e.g. training on MNLI (Williams et al., 2017) and testing on SNLI). Wang et al. (2019) report problems in generalizability when the two pairs are swapped. The idea is that one should expect the same accuracy for contradiction and neutral when the pairs are swapped (neutral remains neutral, and contradiction remains a contradiction[2]), and a lower accuracy for entailment (given that entailment turns neutral when the pairs are swapped).

2 Datasets and experiments

Our experiments are performed on the Multi-Genre Natural Language Inference (MNLI) cor-

[1] The set of punctuation symbols we consider are: ’!"#$%&()*+,-./:;<=>?@[]\^_‘{}|’

[2] Even though one can imagine exotic, non-symmetric definitions of “neutral” and “contradiction”, we are not aware of any syste or dataset using such a definition.

pus (Williams et al., 2017) (and variants thereof, as described below). MNLI consists of 433k human-written sentence pairs labeled with entailment, contradiction and neutral. MNLI contains sentence pairs from ten distinct genres[3] of both written and spoken English. Only five genres are included in the training set. The development and test sets have been divided into matched and mismatched, where the former includes only sentences from the same genres as the training data and the latter include sentences from the remaining genres not present in the training data.

We consider three variants of MNLI:

(*orig*) This variant is the original MNLI with no changes whatsoever.

(p) To obtain this variant we make punctuation consistent throughout examples by adding full stops at the end of each sentence.

($\neg p$) To obtain this variant we remove all non-alphanumeric characters from each sentence. This also remove special characters that are sometimes not classified as punctuation, such as the dollar sign. However, such characters occur so seldom that they have little influence on the results, either way (see Table 1).

Appending a sentence-final stop is in general reasonable, especially for the non-dialogue examples. For the dialogue part of the MNLI dataset, this is unnatural as final stops typically are not expressed in dialogue.

To convey an idea of the amount of data that our transformation impact, we show the raw and relative count[4] of punctuation symbols in Table 1. In total, relative to word-tokens, punctuation symbols account for about 11.5% of the tokens.

SYMBOL	COUNT	%
,	672354	3.544
.	632460	3.334
’	426014	2.246
-	188124	0.992
)	66498	0.351
(	66210	0.349
?	41530	0.219
”	27246	0.144
;	18182	0.096
!	11384	0.060
$	8724	0.046
:	6162	0.033
/	5746	0.030
[	1920	0.010
]	1872	0.010
&	1032	0.005
%	1014	0.005
_	666	0.003
*	186	0.001
@	162	0.001
=	150	0.001
#	114	0.001
+	66	0.0003
‘	24	0.0001
~	12	6.32e-05
\	12	6.32e-05
{	12	6.32e-05

Table 1: Count of punctuation symbols used in the training examples of MNLI.

2.1 Experiments

We perform two sets of experiments:

In the first set, designed to test (H1), we train NLI models for either of the three ($orig$, p, $\neg p$) variants and test on either the p or $\neg p$ variants. Additionally, we train on $orig$ and test on $orig$, as a baseline result.

In the second set, we designed a dataset to test (H2), that is, whether NLI models are able to detect semantically relevant punctuation. This experiment is performed the same way as the first set, but we replace the MNLI test data with our own dataset. The dataset we constructed for this contain a number of problems whose correct label depends on the presence or absence of punctuation. Here are some representative examples (& separates the premise from the hypothesis, label follows in parentheses):

(1) I thank, my mother, Anna, Smith and John & I thank four people (E)

(2) I thank, my mother Anna, Smith and John & I thank two people (C)

(3) The notion of good, god, is incomprehensible & Good is incomprehensible (E)

(4) The notion of good, god, is incomprehensible & Good god is incomprehensible (C)

[3] face to face conversations, telephone ones, letters, oxford university press publications, etc.

[4] Relative to the number of total tokens in the MNLI dataset

The first two examples are cases where the commas are used to denote the conjunction of more than one conjunct. Removing the comma between "my mother" and "Anna" in 2 has a significant effect on counting: what is taken to be two entities in 1, are one in 2. In 3 and 4, we get a different label depending on whether the hypothesis refers to the property "good" (E) or the adjectival modification "good god" (C). The test set consists of 18 examples which can be seen in Table 4.

3 Models

The experiments are performed using three models:

BiLSTM The simplest model is a bidirectional LSTM that encodes the premise and hypothesis, then applies max pooling. The model then concatenates the premise and hypothesis in the standard fashion (Conneau et al., 2017; Talman et al., 2019): $[p; h; p - h; p * h]$ where p is the premise representation and h the hypothesis representation. A three-layer perceptron with leaky ReLU activation between the layers then assigns a class to the example.

HBMP The second model is described by Talman et al. (2019). The model is a three-layer bidirectional LSTM, wherein between the layers a representation is extracted through max pooling. The final representation for each sentence is the concatenation of all intermediate representations $[h_0; h_1, h_2]$. The same representation as with the BiLSTM, $[p; h; p - h; p * h]$ where p and h respectively is the concatenation of all intermediate representations, is then passed to a three-layer perceptron with leaky ReLU activation and dropout.

BERT Our third model is a transformer model, BERT (Devlin et al., 2018). We use the BERT base model from the transformer library (Wolf et al., 2019). To train BERT we use a three layer perceptron with Leaky ReLU activations on top of the BERT model and fine-tune. The BERT model process the premise and hypothesis is parallel and there is no need to explicitly combine them as with the previous models. For the classification of a sentence pair, we use the CLS token generated by BERT that contain information about both sentences.

4 Experimental setup

For each architecture (BERT, HBMP, and BiLSTM) we perform experiments by training four models, two trained and validated on the dataset with punctuation and two models trained and validated on the dataset without punctuation. To asses the effect of our data augmentation we test the model on the other dataset, i.e. a model trained and validated without punctuation is tested on the dataset with punctuation. We measure the performance in terms of accuracy.

For HBMP and the BiLSTM models we use the default hyperparameters reported by Talman et al. (2019) with GloVe (Pennington et al., 2014) word embeddings[5]. The BERT model is fine-tuned with the default model hyperparameters. We use the Adam optimizer with a learning rate of 0.00002 and a batch size of 24.

5 Results

5.1 First experiment set

The results from the first experiment are shown below in Table 2. The experiment shows the accuracy for the models trained on the MNLI variations with and without punctuation and their accuracy on all variations.

MODEL	TEST	MA	MM
BiLSTM$_{orig}$		.724	.723
BiLSTM$_{p}$	p	.723	.724
BiLSTM$_{p}$	$\neg p$	.428	.414
BiLSTM$_{\neg p}$	$\neg p$	.714	.727
BiLSTM$_{\neg p}$	p	.424	.430
HBMP$_{orig}$		.729	.733
HBMP$_{p}$	p	.728	.729
HBMP$_{p}$	$\neg p$	.430	.408
HBMP$_{\neg p}$	$\neg p$	.729	.732
HBMP$_{\neg p}$	p	.436	.427
BERT$_{orig}$		.833	.839
BERT$_{p}$	p	.835	.837
BERT$_{p}$	$\neg p$	.816	.822
BERT$_{\neg p}$	$\neg p$	.819	.820
BERT$_{\neg p}$	p	.830	.833

Table 2: The effect on punctuation on all three models in terms of accuracy of the MNLI dataset. MA indicate the matched and MM the mismatched test split. $original$ is trained on the unaugmented data, p models trained with punctuation and $\neg p$ models trained without punctuation

[5]Trained on 840 billion tokens.

The results indicate that when the RNN-based models are tested on the same dataset as it is trained on, the results are similar to that of the original model. However, when we test on the opposite dataset the performance drops drastically (about 30 percentage points). We see that the drop in accuracy is about the same for both the matched and mismatched test set. In contrast to the RNN-based models, the transformer model only shows a slight drop in accuracy when presented with test data different from its training data.

5.2 Second experiment set

Full results from the second experiment can be found in Table 4, a subset of the examples can be found in Table 3. The experiment shows the predictions by the HBMP and BERT models trained with and without punctuation on our hand-crafted dataset.

5.3 Experiment one analysis

The experiment shows that the BLSTM and HBMP models trained with punctuation drops significantly in accuracy when tested on data without punctuation. This indicates that when removing punctuation the model changes its prediction incorrectly. Most of the removed punctuation does not change the meaning, rather some information irrelevant the the relationship between the two sentences (such as sentence-final stop).

Inspecting the output of the HBMP model we can see that in many cases, removing a sentence-final stop flips the models' prediction. In example (5) and (6), both the model trained on punctuation and the one without fail to predict that the final stop does not add any meaning.

(5) not yourself . & only you . (C)
HBMP$_p$ = C
HBMP$_{\neg p}$ = E
BERT$_p$ = N
BERT$_{\neg p}$ = N

(6) not yourself & only you (C)
HBMP$_p$ = E
HBMP$_{\neg p}$ = C
BERT$_p$ =N
BERT$_{\neg p}$ = N

In examples (7) and (8)[6], the sentence-final stop has been removed, as well as a comma. In such a case, the comma does not add any meaning but acts as a separator of clauses. The removal or addition of this comma flips the prediction of the models. This shows that irrelevant changes both involving commas and sentence-final stops can flip the model's prediction without any semantic motivation.

(7) P = so they set about clearing the land for agriculture , setting fire to massive tracts of forest .
H = as a result , the land was devastated by erosion . (N)
HBMP$_p$ = N
HBMP$_{\neg p}$ = C
BERT$_p$ = N
BERT$_{\neg p}$ = N

(8) P = so they set about clearing the land for agriculture setting fire to massive tracts of forest
H = as a result the land was devastated by erosion (N)
HBMP$_p$ = C
HBMP$_{\neg p}$ = N
BERT$_p$ =E
BERT$_{\neg p}$ = C

BERT assigns the neutral class regardless of punctuation in examples (5) to (7), indicating that the choice of punctuation in training and test does not impact its decision. For example (8) there is no punctuation in the premise and hypothesis, but the different BERT models assign two different classes, *entailment* by the model trained on punctuation and *contradiction* by the model trained without punctuation.

A possible explanation for why the accuracy of BERT does not behave similarly to that of the LSTM based models is that the pretraining of BERT allows the model to better ignore variations in the input. However, the HBMP model also uses pre-trained information in the form of GLoVE vectors, yet we do not see HBMP handling the discrepancy between the training and the test well. Albeit the pre-training of GLoVE and BERT are different, in the essence they are the same. Both model the meaning of words based on their surroundings in the neural architecture. Thus, the relevant difference between the models relevant to the absence or presence of punctuation is whether the model use self-attention or an LSTM to create representations of sentences.

[6]For clarity, the premise is indicated by P and the hypothesis by H.

n	Premise	Hypothesis	Gold	Pred	Model
0	I thank, my mother, Anna, Smith and John	I thank four people	E	N	HBMP$_{\neg p}$
1	I thank, my mother, Anna Smith and John	I thank three people	E	N	HBMP$_{\neg p}$
8	I hear John says 'come here'	I hear John speaking	E	E	HBMP$_{\neg p}$
9	I hear 'John says come here'	I hear John speaking	C	N	HBMP$_{\neg p}$
14	No, god is good	God is good	E	E	HBMP$_{\neg p}$
15	No god is good	There is no good god	E	E	HBMP$_{\neg p}$
16	No, god is good	There is no good god	C	E	HBMP$_{\neg p}$
17	No god is good	God is good	C	C	HBMP$_{\neg p}$
0	I thank, my mother, Anna, Smith and John	I thank four people	E	E	HBMP$_{p}$
1	I thank, my mother, Anna Smith and John	I thank three people	E	E	HBMP$_{p}$
8	I hear John says 'come here'	I hear John speaking	E	E	HBMP$_{p}$
9	I hear 'John says come here'	I hear John speaking	C	E	HBMP$_{p}$
14	No, god is good	God is good	E	E	HBMP$_{p}$
15	No god is good	There is no good god	E	E	HBMP$_{p}$
16	No, god is good	There is no good god	C	E	HBMP$_{p}$
17	No god is good	God is good	C	C	HBMP$_{p}$
0	I thank, my mother, Anna, Smith and John	I thank four people	E	E	BERT$_{\neg p}$
1	I thank, my mother, Anna Smith and John	I thank three people	E	C	BERT$_{\neg p}$
8	I hear John says 'come here'	I hear John speaking	E	C	BERT$_{\neg p}$
9	I hear 'John says come here'	I hear John speaking	C	E	BERT$_{\neg p}$
14	No, god is good	God is good	E	E	BERT$_{\neg p}$
15	No god is good	There is no good god	E	E	BERT$_{\neg p}$
16	No, god is good	There is no good god	C	E	BERT$_{\neg p}$
17	No god is good	God is good	C	E	BERT$_{\neg p}$
0	I thank, my mother, Anna, Smith and John	I thank four people	E	E	BERT$_{p}$
1	I thank, my mother, Anna Smith and John	I thank three people	E	C	BERT$_{p}$
8	I hear John says 'come here'	I hear John speaking	E	C	BERT$_{p}$
9	I hear 'John says come here'	I hear John speaking	C	E	BERT$_{p}$
14	No, god is good	God is good	E	E	BERT$_{p}$
15	No god is good	There is no good god	E	E	BERT$_{p}$
16	No, god is good	There is no good god	C	E	BERT$_{p}$
17	No god is good	God is good	C	E	BERT$_{p}$

Table 3: Results on a subset of the examples in our constructed dataset. E is entailment, N is neutral and C is contradiction. The model column indicate which HBMP model configuration was used (trained with punctuation p, or without $\neg p$).

From this, we pose a tentative hypothesis that self-attention more easily learn to ignore irrelevant input tokens for a task than the LSTM. However, to confirm this we need to perform more expensive experiments.

5.4 Experiment two analysis

None of the models perform very well for this dataset. The HBMP$_p$ model has an accuracy of 61.1% while the HBMP$_{\neg p}$ has an accuracy of 44.4%. The BERT$_p$ model has an accuracy of 44.4% while the BERT$_{\neg p}$ has an accuracy of 38.8%.

For example, both models are tricked by comma removal in (2). An interesting case involves cases where the comma is removed from "No, god" turning it into a negative quantifier "no god". The models are tricked when asked to infer "There is no good god" from "No, god is good" (they predict E instead of C). Another example where the models are tricked by comma removal is when listing items. In the example "I thank, my mother, Anna Smith and John" there are three entities being thanked. The comma placement indicates that "Anna Smith" is one person, and not two. Only HBMP$_p$ successfully predicts that "I thank three people" is an entailment for this example. The quotation examples are also challenging. Both systems are tricked when they are asked to judge whether "I hear John speaking" follows: a) from "I hear John says 'come here' ", and b) "I hear 'John says come here' ". Both models correctly predict a) but fail on b). However, they give a different wrong label, (N) for HBMP$_{\neg p}$ and (E) for HBMP$_p$.

6 Conclusions

The conclusions of this paper can be summarized as follows:

Only BERT is robust to irrelevant changes in punctuation (H1 is validated for BERT). The other models see a significant drop in performance when for any mismatch of the presence of punctuation between training and testing sets. However, the

presence or absence of the full stop at the end of a sentence has little effect.

This statement rests on the observation that punctuation is generally semantically insignificant in MNLI. This fact has not been tested using a model but rather relies on manual inspection of the data.

We have evidence that no model is capable of taking into account cases where punctuation is meaningful. At this stage of our research, this evidence does not rely on a large body of data. This result is not surprising because of the above observation (namely, there is not enough meaningful punctuation in the training set). Yet, we use pre-trained embeddings (BERT) which have been trained on very large dataset, and it could not be ruled out *a priori* that such embeddings did not contain information related to the meaning of punctuation.

As a general remark, it seems to us useful, if not necessary, to extend the present datasets for NLI to include examples where punctuation is actually meaningful. In general, this is part of a discussion of extending current datasets to include cases of inference where more fined-grained phenomena are taken into consideration Chatzikyriakidis et al. (2017); Bernardy and Chatzikyriakidis (2019, 2020). This also connects with the generalization capabilities of NLI models that were briefly brought up in the introduction. However, the goal should not only be to create many diverse datasets that can get very fine-grained for numerous syntactic phenomena. What we further need are models that will have the ability to generalize well to new data after they have been trained on datasets that represent a much more diverse and rich picture of NLI, and are not prone to similar problems as these have been reported in the literature (Glockner et al., 2018; Talman and Chatzikyriakidis, 2018; Wang et al., 2019; Poliak et al., 2018).

7 Future work

In future work, we plan to continue pursuing the question of model generalizability by investigating how neural models for natural language inference can be adapted to take into account fine-grained semantic phenomena. More specifically, how can models be adapted to learn what constitutes a meaningful part of a sentence, in terms of semantics, and what is not meaningful. We can notice that the phenomena of punctuation is primarily "syntactic sugar", by constructing a sentence in a certain way syntactically (by inserting or removing punctuation). To exploit this we plan to incorporate syntactic representations of sentences.

Acknowledgments

The research reported in this paper was supported by grant 2014-39 from the Swedish Research Council, which funds the Centre for Linguistic Theory and Studies in Probability (CLASP) in the Department of Philosophy, Linguistics, and Theory of Science at the University of Gothenburg.

References

Jean-Philippe Bernardy and Stergios Chatzikyriakidis. 2019. What kind of natural language inference are nlp systems learning: Is this enough? In *13th International Conference on Agents and Artificial Intelligence*.

Jean-Philippe Bernardy and Stergios Chatzikyriakidis. 2020. Improving the precision of natural textual entailment problem datasets. In *Proceeding of LREC 2020*.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Stergios Chatzikyriakidis, Robin Cooper, Simon Dobnik, and Staffan Larsson. 2017. An overview of natural language inference data collection: The way forward? In *Proceedings of the Computing Natural Language Inference Workshop*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

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*.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences. *arXiv preprint arXiv:1805.02266*.

Jeffrey Pennington, Richard Socher, and Christopher D 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.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*.

Aarne Talman and Stergios Chatzikyriakidis. 2018. Testing the generalization power of neural network models across nli benchmarks. *arXiv preprint arXiv:1810.09774*.

Aarne Talman, Anssi Yli-Jyrä, and Jörg Tiedemann. 2019. Sentence embeddings in nli with iterative refinement encoders. *Natural Language Engineering*, 25(4):467–482.

Haohan Wang, Da Sun, and Eric P Xing. 2019. What if we simply swap the two text fragments? a straightforward yet effective way to test the robustness of methods to confounding signals in nature language inference tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7136–7143.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

n	Premise	Hypothesis	Gold	Pred	Model
0	I thank, my mother, Anna, Smith and John	I thank four people	E	N	HBMP$\neg p$
1	I thank, my mother, Anna Smith and John	I thank three people	E	N	HBMP$\neg p$
2	I thank, my mother Anna, Smith and John	I thank two people	C	E	HBMP$\neg p$
3	I thank, my mother Anna Smith and John	I thank three people	C	E	HBMP$\neg p$
4	I thank, my mother Anna, Smith and John	I thank more than two people	E	N	HBMP$\neg p$
5	I thank my mother Anna, Smith and John	My mother is called Anna Smith	N	N	HBMP$\neg p$
6	I thank my mother, Anna Smith and John	My mother is called Anna Smith	N	N	HBMP$\neg p$
7	I thank my mother Anna Smith and John	My mother is called Anna Smith	E	E	HBMP$\neg p$
8	I hear John says 'come here'	I hear John speaking	E	E	HBMP$\neg p$
9	I hear 'John says come here'	I hear John speaking	C	N	HBMP$\neg p$
10	The notion of good, god, is incomprehensible	Good is incomprehensible	E	N	HBMP$\neg p$
11	The notion of good god is incomprehensible	Good god is incomprehensible	E	E	HBMP$\neg p$
12	The notion of good, god, is incomprehensible	Good god is incomprehensible	C	E	HBMP$\neg p$
13	The notion of good god is incomprehensible	Good is incomprehensible	N	C	HBMP$\neg p$
14	No, god is good	God is good	E	E	HBMP$\neg p$
15	No god is good	There is no good god	E	E	HBMP$\neg p$
16	No, god is good	There is no good god	C	E	HBMP$\neg p$
17	No god is good	God is good	C	C	HBMP$\neg p$
0	I thank, my mother, Anna, Smith and John	I thank four people	E	E	HBMPp
1	I thank, my mother, Anna Smith and John	I thank three people	E	E	HBMPp
2	I thank, my mother Anna, Smith and John	I thank two people	C	E	HBMPp
3	I thank, my mother Anna Smith and John	I thank three people	C	E	HBMPp
4	I thank, my mother Anna, Smith and John	I thank more than two people	E	N	HBMPp
5	I thank my mother Anna, Smith and John	My mother is called Anna Smith	N	N	HBMPp
6	I thank my mother, Anna Smith and John	My mother is called Anna Smith	N	N	HBMPp
7	I thank my mother Anna Smith and John	My mother is called Anna Smith	E	E	HBMPp
8	I hear John says 'come here'	I hear John speaking	E	E	HBMPp
9	I hear 'John says come here'	I hear John speaking	C	E	HBMPp
10	The notion of good, god, is incomprehensible	Good is incomprehensible	E	E	HBMPp
11	The notion of good god is incomprehensible	Good god is incomprehensible	E	E	HBMPp
12	The notion of good, god, is incomprehensible	Good god is incomprehensible	C	E	HBMPp
13	The notion of good god is incomprehensible	Good is incomprehensible	N	C	HBMPp
14	No, god is good	God is good	E	E	HBMPp
15	No god is good	There is no good god	E	E	HBMPp
16	No, god is good	There is no good god	C	E	HBMPp
17	No god is good	God is good	C	C	HBMPp
0	I thank, my mother, Anna, Smith and John	I thank four people	E	E	BERT$\neg p$
1	I thank, my mother, Anna Smith and John	I thank three people	E	C	BERT$\neg p$
2	I thank, my mother Anna, Smith and John	I thank two people	C	E	BERT$\neg p$
3	I thank, my mother Anna Smith and John	I thank three people	C	E	BERT$\neg p$
4	I thank, my mother Anna, Smith and John	I thank more than two people	E	E	BERT$\neg p$
5	I thank my mother Anna, Smith and John	My mother is called Anna Smith	N	E	BERT$\neg p$
6	I thank my mother, Anna Smith and John	My mother is called Anna Smith	N	E	BERT$\neg p$
7	I thank my mother Anna Smith and John	My mother is called Anna Smith	E	E	BERT$\neg p$
8	I hear John says 'come here'	I hear John speaking	E	C	BERT$\neg p$
9	I hear 'John says come here'	I hear John speaking	C	E	BERT$\neg p$
10	The notion of good, god, is incomprehensible	Good is incomprehensible	E	E	BERT$\neg p$
11	The notion of good god is incomprehensible	Good god is incomprehensible	E	E	BERT$\neg p$
12	The notion of good, god, is incomprehensible	Good god is incomprehensible	C	E	BERT$\neg p$
13	The notion of good god is incomprehensible	Good is incomprehensible	N	E	BERT$\neg p$
14	No, god is good	God is good	E	E	BERT$\neg p$
15	No god is good	There is no good god	E	E	BERT$\neg p$
16	No, god is good	There is no good god	C	E	BERT$\neg p$
17	No god is good	God is good	C	E	BERT$\neg p$
0	I thank, my mother, Anna, Smith and John	I thank four people	E	E	BERTp
1	I thank, my mother, Anna Smith and John	I thank three people	E	C	BERTp
2	I thank, my mother Anna, Smith and John	I thank two people	C	E	BERTp
3	I thank, my mother Anna Smith and John	I thank three people	C	E	BERTp
4	I thank, my mother Anna, Smith and John	I thank more than two people	E	E	BERTp
5	I thank my mother Anna, Smith and John	My mother is called Anna Smith	N	E	BERTp
6	I thank my mother, Anna Smith and John	My mother is called Anna Smith	N	E	BERTp
7	I thank my mother Anna Smith and John	My mother is called Anna Smith	E	E	BERTp
8	I hear John says 'come here'	I hear John speaking	E	C	BERTp
9	I hear 'John says come here'	I hear John speaking	C	E	BERTp
10	The notion of good, god, is incomprehensible	Good is incomprehensible	E	E	BERTp
11	The notion of good god is incomprehensible	Good god is incomprehensible	E	E	BERTp
12	The notion of good, god, is incomprehensible	Good god is incomprehensible	C	E	BERTp
13	The notion of good god is incomprehensible	Good is incomprehensible	N	E	BERTp
14	No, god is good	God is good	E	E	BERTp
15	No god is good	There is no good god	E	E	BERTp
16	No, god is good	There is no good god	C	E	BERTp
17	No god is good	God is good	C	E	BERTp

Table 4: Constructed dataset. E is entailment, N is neutral and C is contradiction. The Model column indicate which model was used (trained with punctuation p, or without $\neg p$).