Natural Language Processing (NLP) Project Report

1. Introduction

Sequence classification is a common problem in the domain of NLP, such as POS tagging, information retrieval, abnormal detection and incremental parsing. A large number of different model frameworks are able to deal with this problem, for example, Logistic Regression, Recurrent Neural Network (RNN) and Transformers. Logistic and Fasttext models are usually sufficient enough to handle simple classification problems, for instance, emotional semantic analysis and fake news detection. However, there are some challenging tasks and dataset, even the pre-trained Transformers cannot handle very well. One example of this is the GLUE (General Language Understanding Evaluation) -WNLI (Winograd Natural Language Inference) dataset. The GLUE-WNLI dataset was constructed in the form of sequence classification and its task was to handle pronouns reference resolutions and language reasoning. Considering the challenges of this dataset's reference disambiguate resolution, it would be worthwhile to experiment different model architectures, compare their performances, and analyse the reference resolution challenges on GLUE-WNLI.

In this report, we experimented with Logistic Regression, RNN-GRU (with and without Glove Embeddings) and pre-trained Transformer models on the sequence classification dataset of GLUE-WNLI. We then compared different model architectures' performances on the pronouns reference disambiguation task. Finally, we discussed this task's challenges and some related work.

2. Dataset and Problems

2.1 Dataset Introduction

GLUE is a collection of data for training, evaluating, and analysing natural language understanding systems. It consists of different tasks of various difficulty levels. Among this collection, the Winograd Schema Challenge (Levesque et al. 2011) is a reading comprehension task, and there are pairs of sentences that differ in only one or two words. Such pairs of sentences contain an ambiguity that is caused by the pronouns in the first sentence. The task for the machine is to resolve the ambiguity with the external world knowledge and reasoning about pronoun reference resolution. For example, one pair of sentences is: (1. I stuck a pin through a carrot. When I pulled the pin out, it had a hole. 2. The carrot had a hole.) and the label for this pair is "1", since the "it" in the sentence 1 refers to the "carrot". Thus this pair of sentences expressed the same meanings and the reasoning about the pronoun "it" was correct. As we can see from this example, this task is essentially a pronoun reference resolution task.

While Winograd converted the problem into sentence pair classification. In fact, he constructed sentence pairs by replacing the ambiguous pronoun with each possible referent. Therefore the task was converted to the prediction of whether the sentence with the pronoun substituted is entailed by the original sentence and thus to tell whether the hypothesis follows from the premise. In general, pronoun reference resolution can be handled by hard agreement constraints and heuristics such as recency. However, Winograd's intention was that selectional restrictions could not handle ambiguity (Levesque et al. 2012). Besides, according to the Winograd Schemas, there should be no obvious statistical test over the text corpora that will reliably disambiguate these correctly, which means there is no simple statistical method that could be applied to resolve the ambiguity. This dataset is regarded as a challenge for NLP AI tasks. The advantages for this dataset is that:

- Classification is clear-cut, binary-choice.
- No expert external knowledge is needed to reason the pronoun reference ambiguity.
- It is difficult and beyond the most state of art models, while it is very easy for humans to disambiguate the pronoun reference.

2.2 Selection criterias

Dataset Size
This dataset consists of three sections: train, validation and test. All of them are relatively small, which means it is unnecessary to adapt GPU's architecture for transformer models. Considering the time and computation resources limitations for this assignment, dataset size is the most important consideration. During our initial experiments on Squad1.1 with Bert transformer models, just the set up for a proper CUDA environment and GPUs embedded in Pytorch took a significantly long time. Even while we had started the training process, because of the large size of SQUAD1.1, the training process would take 6 days to finish. As a result, we had to re-source on another size-proper dataset. We went back and inspected through the GLUE dataset and finally decided on WNLI due to its smaller size.

Dataset Labels
As stated above, the labels for this dataset are binary: 1 and 0. This was the second important reason why we chose this dataset. As a traditional NLP task, almost all the models are able to handle classification tasks. For this assignment, we thought it would be meaningful to run different models that we have learned in the course, specifically Logistic, RNN, and transformers. In addition, a benefit of choosing this dataset is that we have a good source to compare and analyse the different models based on the characteristics of WNLI.

Dataset Characteristics
WNLI task includes both commonsense reasoning and natural language understanding, which was introduced as an alternative to the Turing test (Kocijan et al. 2020). This task necessarily requires common sense knowledge to resolve correctly. All the sentence pairs disambiguate can be easily solved by humans, while it is hard for machines, which requires a deep understanding of the content of the text and the situation it describes.

Evaluation
The evaluation benchmark for WNLI is accuracy or F1 measure. This is because the training set is balanced between two class labels. However, since there are no test labels attached in the dataset, we could only use the validation dataset to compare among different models. Once we picked the best performance model, we could upload the model to test under the official benchmark. As we did some initial experiments on the Logistic model, the performance is nearly random, and the confusion matrix would be used to denote models' performances as well. As a result, this evaluation method would be consistent among different model architectures.

3. Modelling and Analysis

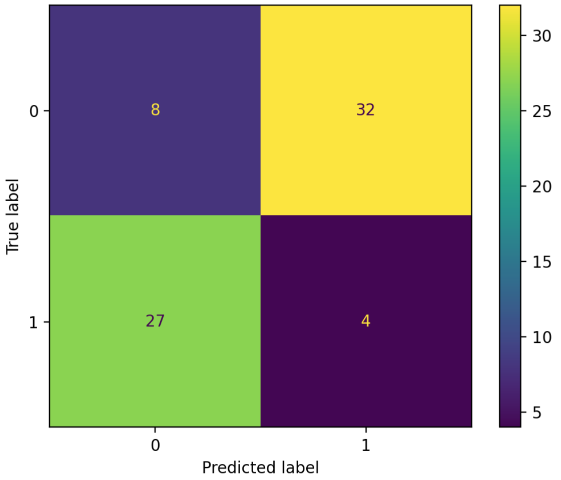
In general, sequence classification tasks are very sensitive to the words' position in the sequences, which is the main difference from general classification using Bag of Words vectors (TF-Idf). Specifically, Bag of Words ignores the word position and the context information for the current word, which is crucial to disambiguate pronoun reference in the WNLI dataset. As for Fasttext, because of the window size of Continuous Bag of Words (CBOW) or Skip-gram, it is still difficult for the model to capture the whole semantic meanings between the sentence pairs. This means that it remains difficult for the Fasttext model to learn to disambiguate pronoun references.

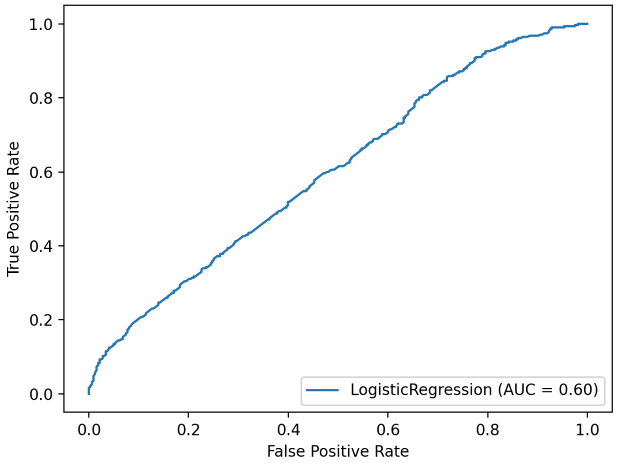
Even though RNN models take word positions and sequence context into consideration, their limitation is on the longer sequence, especially for whose sequence size is in the hundreds. As for transformer models, the pre-training process has embedded external language knowledge into the models, which is exactly the key to pronoun disambiguate resolution. As stated above, WNLI task is unambiguous for humans because of humans' external language knowledge. Considering this characteristic of pre-trained transformer models, our hypothesis is that these models would perform better. In this section we train different models on the WNLI dataset to compare and analyse their performances.

3.1 Logistic Regression

The first model is a linear Logistic model, we chose TF-Idf to denote the dataset features. The implementation python file *'train_logistic_classifier'* was included with the assignment file. One finding during experiments on Logistics is that max features greatly determined the results. Specifically, since the vocabulary size of the training dataset is 1504 and of validation is 532, the max-feature of TF-Idf was set to 2000. This is because we want to include all the words, however the training accuracy was still not high. While when we set the max-features to 200, the validation accuracy improved.

Precision	Recall	Training Accuracy	Validation Accuracy
0.111	0.129	0.613	0.197



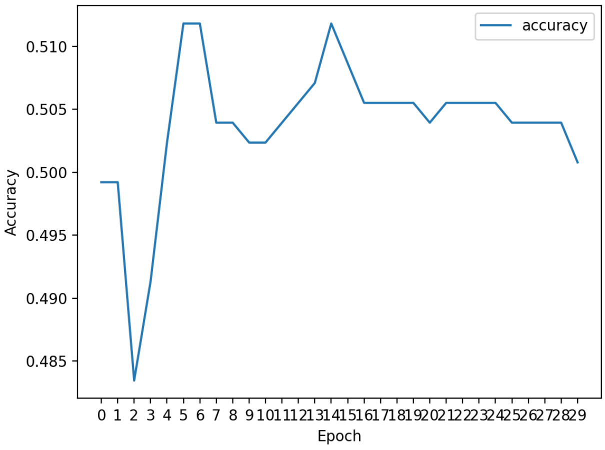


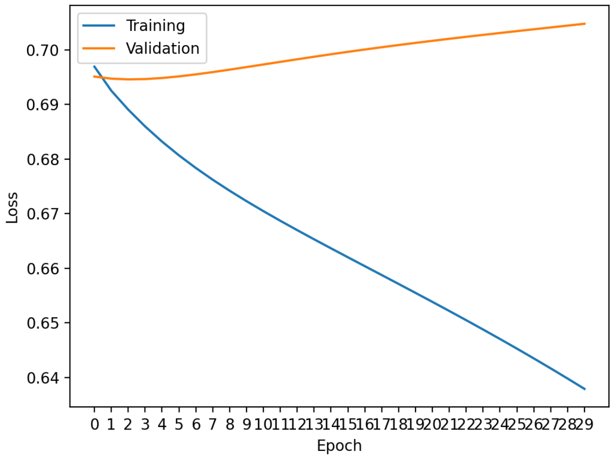
As we can see from the validation accuracy, this Logistics model performed quite bad on this task. Besides according to the confusion-matrix above, we can see the prediction on the validation dataset was bad too since for both 1 and 0 classes, this model predicted falsely for the majority labels. The low training accuracy denoted that this task was not linear, more complicated models could be able to capture the non-linear patterns. According to the ROC-curve, we can infer that this Logistic model performed quite close to randomly.

One important conclusion is that this Logistic model fell in the trap of WNLI challenging schema. Because the class labels are balanced in this dataset, theoretically, training accuracy was supposed to be close to validation accuracy. However, the validation accuracy here was quite low, which exactly indicated this model remembered the training data and thus fell in the adversarial trap: hypotheses are sometimes shared between training and validation examples, so if a model memorizes the training examples, it will predict the wrong label on corresponding validation set example. As we can see from here, WNLI is very challenging in this aspect.

3.2 RNN-GRU

We implemented a RNN-GRU model by using the python file *'train_gru'* included in the assignment file. After experimenting on several different parameters, this setting performed quite well and stable: hidden dimension is 64, learning rate is *2e-5*, and epoch is 30. The loss-plotting and accuracy-plotting were shown below:





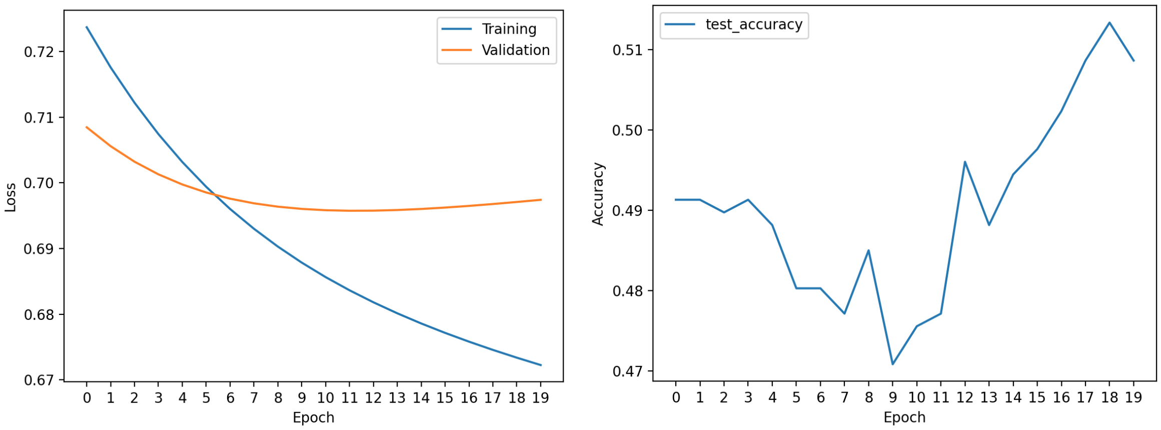
According to these plotting graphs, we can find the validation accuracy started to drop around epoch 28 and the loss for validation started to increase around epoch 28. We can infer that from epoch 25, over-fitting issues start to arise.

The other interesting finding is that validation loss was nearly keeping the same during the process. We can infer that this GRU model did not improve the learning as with the increasing epochs. The model was learning the fitting noise from the training data or memorising the training data, while still was not able to predict properly on the validation dataset. This behavior exactly responded to the fact, WNLI is very challenging for machines to disambiguate pronoun reference. This means the external language knowledge is necessary for machines to perform this task. Therefore, the following model adapted pretrained Glove Embeddings to represent sentence pairs.

3.3 RNN-GRU with Glove Embeddings

Considering the necessity of external language knowledge of this dataset task, rather than training word embedding from scratch, we adapted Glove pre trained embeddings to leverage the external knowledge for this GRU model. We used "Wikipedia 2014 + Gigaword 5" which is the smallest file ("glove.6B.zip") with 822 MB. It was trained on a corpus of 6 billion tokens and contains a vocabulary of 400 thousand tokens. We picked the smallest one with words represented by vector dimension of 50 ("glove.6B.50d.txt"). Glove used the ratio of co-occurrence probabilities between words to denote the correlation between words.

However, Glove still lacks enough context information to represent diverse semantics. Moreover, word embedding could not express diverse word senses. WNLI dataset specifically used adversarial tricks between training and validation data, which required the model to capture different semantics in different contexts for the same words. Nonetheless, word embedding lacks the ability to represent diverse word senses, thus this GRU model still fell in the adversarial trap between training and validation. As we can notice from the results, the accuracy just improved around 15%, while the validation loss decreased during epochs. Therefore the performance was better than GRU without Glove. (Note: since "glove.6B.50d.txt" is too large to upload with assignment file, we did not upload this file and running "train_externalgru.py" requires to download the Glove data and put in the right directory specified in path parameters of python code file.)



3.4 Pre-trained Transformers

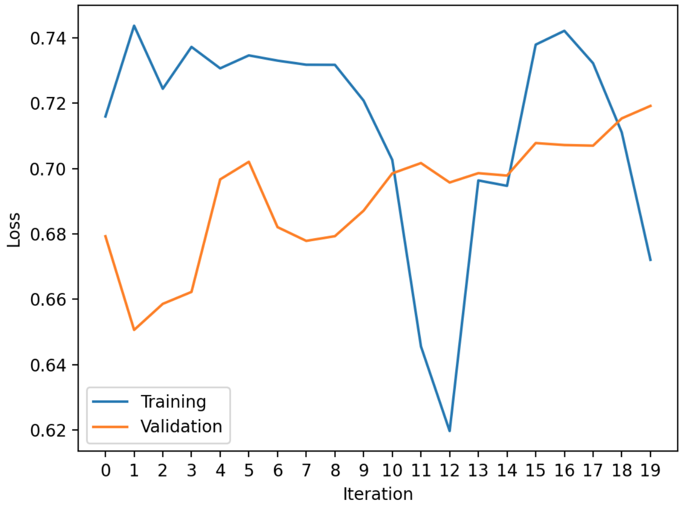
Unlike the Logistic-regression and RNN-GRU models experimented above, pre-trained transformer models have been trained on massive datasets, so such models have learnt external language knowledge, which is exactly the key to solving the WNLI task. However different model architectures and pertaining tend to have different advantages among different NLP tasks. Therefore, it is important for us to understand how different pre-trained models' architectures and what types of tasks they tend to perform better. In this section, we experimented with BERT, XLM, XLNet, and RoBERTa models and analysed their performance on WNLI, therefore comparing among different models.

3.4.1 BERT-base-uncased

BERT is the first pre-trained model that uses bidirectional language representation, which helps the model understand the context, which is crucial to the WNLI dataset. However this model was pre-trained on fairly small-size plain text corpus, the BooksCorpus and English Wikipedia, it is unlikely to have learned enough language knowledge to perform very well on challenging tasks like WNLI (Devlin et al. 2018). In addition, even in the original paper of the BERT model, Devlin et al. (2018) excluded the WNLI from other GLUE dataset, which exactly denotes why WNLI is very challenging to resolve. We fine-tuned Bert-base-uncased on WNLI and the performance was:

Learning rate	Validation loss	Validation accuracy	Epoch number

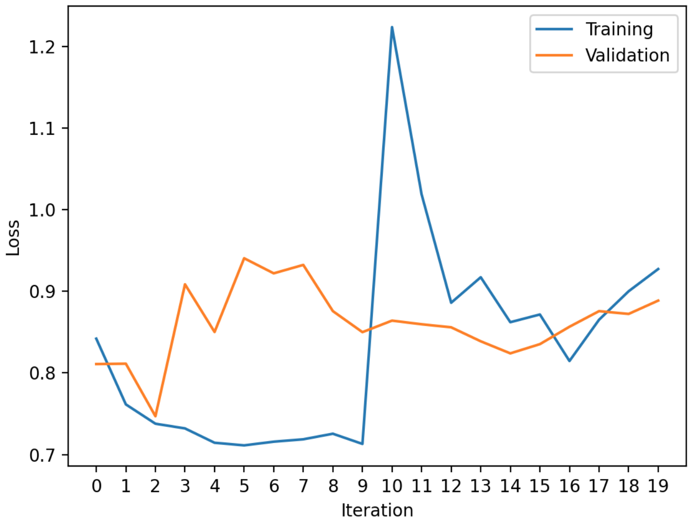
2e-5	0.687	0.567	2



3.4.2 XLM

XLM was developed as an improved version of BERT that allows vocabulary and language knowledge to be shared among multiple languages. XLM was trained with samples composed of the same text in two languages, which allowed the model to learn the context from one and predict words in the other. In addition, as the enhanced version of BERT, XLM was trained on both single language modelling (MLM) and dual-language modelling (TLM) (Lample & Conneau 2019). The model was pre-trained using the XNLI dataset, which includes 112,500 sentence pairs in 15 different languages, and the pre-trained task was to classify the relationship between sentences (Conneau et al. 2018; Lample & Conneau 2019). On the other hand, the model was also trained on the task of casual language modelling (CLM), which is to find the probability of a word given the previous words in a sentence. During pre-training, both MLM and CLM tasks were trained on sequences of more than two consecutive sentences (Lample & Conneau 2019). This means that although the WNLI dataset is only in one language, we think XLM would still be useful as it was pre-trained to learn and classify the contexts and relationships between sentences, which means that it could have more general language knowledge to perform well on the WNLI set.

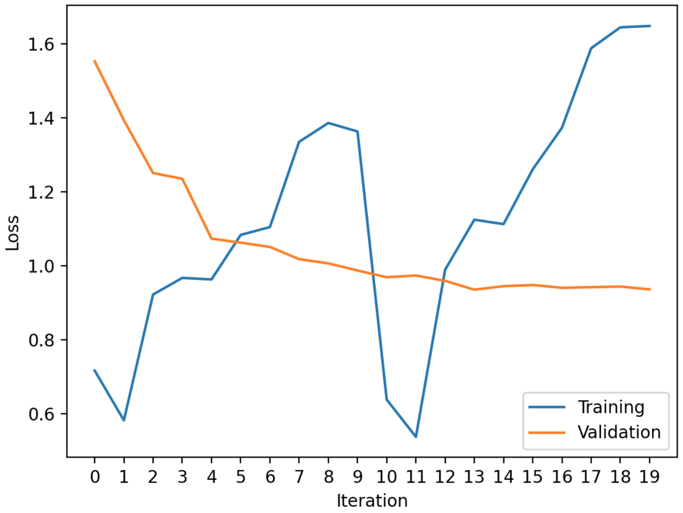
Learning rate	Validation loss	Validation accuracy	Epoch number
1e-4	0.786	0.556	3



3.4.3 XLNet-base-cased

As another version of improved BERT, XLNet focuses on improving and fixing different aspects of BERT compared to XLM. For example, XLNet uses the expected log likelihood of a sequence with respect to all possible permutations of the word order, which makes the tokens on both sides of a word its context. This means that at any position, a word would capture its bidirectional context. Moreover, XLNet was pre-trained on much larger datasets, including the datasets that BERT was trained on (Yang et al., 2019). With this more sophisticated bidirectional context learning and larger pre-training datasets, we think XLNet could achieve higher performance than BERT.

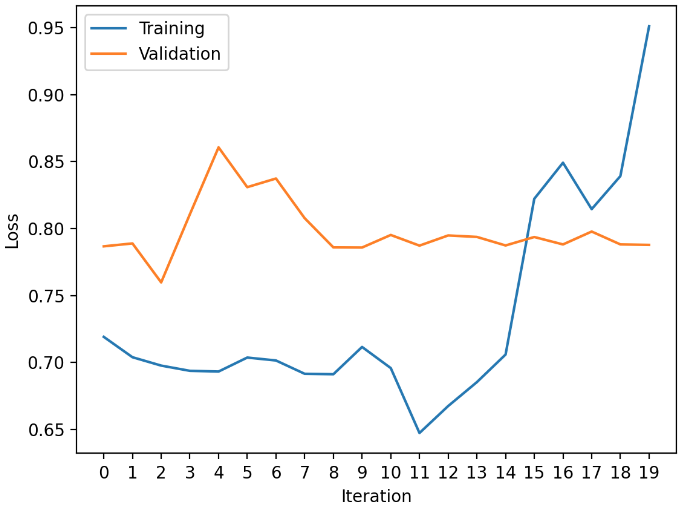
Learning rate	Validation loss	Validation accuracy	Epoch number
1e-4	0.69	0.563	2



3.4.4 RoBERTa-base

Although a reformatted WNLI data was used, out of all the pre-trained models discussed in this paper RoBERTa is the only model that was tested on WNLI. Being yet another enhanced BERT model, there are four main differences that made RoBERTa. First of all, like XLNet, RoBERTa was trained on a significantly larger dataset than BERT. Secondly, RoBERTa uses dynamic masking rather than static masking, which was found to increase performance. Thirdly, during pre-training, inputs are full sentences from one or more documents without the next sentence prediction objective used in BERT. This leads to the last difference, which is that RoBERTa was trained on longer sequences (Liu et al. 2019). We expect RoBERTa's pre-training with multiple full sentences and larger datasets would lead to an improved performance on the WNLI dataset.

Learning rate	Validation loss	Validation accuracy	Epoch number
2e-5	0.756	0.582	3



4. Discussion and Related Work

There are three categories of approaches in solving WNLI: feature-based approaches, neural approaches without pre-training, and pre-trained language models (Kocijan et al. 2020). Feature-based approaches extract semantic information, common sense knowledge, web searches enquiries, and word co-occurrences, which usually are rule-based with logics and discrete optimization algorithms. A disadvantage for such approaches is that they are usually not able to extract relevant information in a sentence. This is crucial for the WNLI dataset, due to the nature of WNLI, even slight noise could negatively affect the predictions (Kocijan et al. 2020). In our experiment models, Logistics with TF-Idf could be treated as one feature-based method, and its low accuracy of prediction exactly demonstrated by Kocijan et al. (2020).

The neural approaches without pre-training can read full sentences, which could overcome the information extraction issue that feature-based approaches suffer from. Kocijan et al. (2020) found the main problem with these approaches is that they lack reasoning capabilities. These neural approaches are most often pre-trained on unstructured text, and they like to utilise the semantic similarities from word embeddings or local context encoded by recurrent neural networks. However, these semantic similarities and local contexts are insufficient for the models to obtain enough information for WNLI (Kocijan et al. 2020). Among our experiment models, RNN-GRU-with Glove could be treated as a neural category, and our result accuracy justified Kocijan's conclusion as well.

The last group of approaches, pre-trained language models, have been achieving the best performances on WNLI compared to the previous two. These are large-scale pre-trained language models that have been pre-trained on large amounts of text datasets, and many of them have even been fine-tuned on WNLI type dataset to improve their performance. In addition, BERT have been used by many researchers for their attempts at tackling WNLI (Kocijan et al. 2020). However, WNLI is still very challenging to most state of art models, and our experiments Transformers did not perform more than 0.65 accuracy. Ruan et al. (2019) utilised BERT's next sentence prediction feature. They trained the language model by dividing a sentence into two, replacing the target pronoun, and have the model predict if the semantics of the second part follows the first. With this method combined with BERT-large, the model achieved 71.1% on WSC273 dataset (Kocijan et al. 2020; Ruan et al. 2019). Moreover, with more training data, Sakaguchi et al. (2020) used RoBERTa and achieved accuracy of 90.1% on WSC273 and 85.6% on WNLI, while the benchmark accuracy originally published was only 65.1% (Kocijan et al. 2020; Wang et al. 2018). However, although these approaches have somewhat successfully tackled the pronoun disambiguation task in short texts, Kocijan et al. (2020) believe that this is yet to be enough for these language models to answer questions involving common sense or everyday situations.

Conneau, A., Lample, G., Rinott, R., Williams, A., Bowman, S.R., Schwenk, H. and Stoyanov, V., 2018. XNLI: Evaluating cross-lingual sentence representations. arXiv preprint arXiv:1809.05053.

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Ju, Y., Zhao, F., Chen, S., Zheng, B., Yang, X. and Liu, Y., 2019. Technical report on conversational question answering. arXiv preprint arXiv:1909.10772.

Kocijan, V., Lukasiewicz, T., Davis, E., Marcus, G. and Morgenstern, L., 2020. A Review of Winograd Schema Challenge Datasets and Approaches. arXiv preprint arXiv:2004.13831.

Lample, G. and Conneau, A., 2019. Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291.

Levesque, H., Davis, E. and Morgenstern, L., 2012, May. The winograd schema challenge. In Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Ruan, Y.P., Zhu, X., Ling, Z.H., Shi, Z., Liu, Q. and Wei, S., 2019. Exploring unsupervised pretraining and sentence structure modelling for winograd schema challenge. arXiv preprint arXiv:1904.09705.

Trinh, T.H. and Le, Q.V., 2018. A simple method for commonsense reasoning. arXiv preprint arXiv:1806.02847.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O. and Bowman, S.R., 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R. and Le, Q.V., 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In Advances in neural information processing systems (pp. 5753-5763).

Y.-P. Ruan, X. Zhu, Z.-H. Ling, Z. Shi, Q. Liu, and S. Wei. Exploring unsupervised pre-training and sentence structure modelling for Winograd Schema Challenge. arXiv:1904.09705, 2019.