A Surprisingly Robust Trick for the Winograd Schema Challenge

Vid Kocijan[1], Ana-Maria Cretu[2], Oana-Maria Camburu[1], Yordan Yordanov[1], Thomas Lukasiewicz[1,3]
[1]University of Oxford
[2]Imperial College London
[3]Alan Turing Institute, London
firstname.lastname@cs.ox.ac.uk, a.cretu@imperial.ac.uk

arXiv:1905.06290v1 [cs.CL] 15 May 2019

Abstract

The Winograd Schema Challenge (WSC) dataset WSC273 and its inference counterpart WNLI are popular benchmarks for natural language understanding and commonsense reasoning. In this paper, we show that the performance of three language models on WSC273 strongly improves when fine-tuned on a similar pronoun disambiguation problem dataset (denoted WSCR). We additionally generate a large unsupervised WSC-like dataset. By fine-tuning the BERT language model both on the introduced and on the WSCR dataset, we achieve overall accuracies of 72.2% and 71.9% on WSC273 and WNLI, improving the previous state-of-the-art solutions by 8.5% and 6.8%, respectively. Furthermore, our fine-tuned models are also consistently more robust on the "complex" subsets of WSC273, introduced by Trichelair et al. (2018).

1 Introduction

The Winograd Schema Challenge (WSC) (Levesque et al., 2012, 2011) was introduced for testing AI agents for commonsense knowledge. Here, we refer to the most popular collection of such sentences as WSC273, to avoid confusion with other slightly modified datasets, such as PDP60 (Davis et al., 2017) and the Definite Pronoun Resolution dataset (Rahman and Ng, 2012), denoted WSCR in the sequel. WSC273 consists of 273 instances of the pronoun disambiguation problem (PDP) (Morgenstern et al., 2016). Each is a sentence (or two) with a pronoun referring to one of the two or more nouns; the goal is to predict the correct one. The task is challenging, since WSC examples are constructed to require human-like commonsense knowledge and reasoning. The best known solutions use deep learning with an accuracy of 63.7% (Opitz and Frank, 2018; Trinh and Le, 2018). The problem is difficult to solve not only because of the commonsense reasoning challenge, but also due to the small existing datasets making it difficult to train neural networks directly on the task.

Neural networks have proven highly effective in natural language processing (NLP) tasks, outperforming other machine learning methods and even matching human performance (Hassan et al., 2018; Nangia and Bowman, 2018). However, supervised models require many per-task annotated training examples for a good performance. For tasks with scarce data, transfer learning is often applied (Howard and Ruder, 2018; Johnson and Zhang, 2017), i.e., a model that is already trained on one NLP task is used as a starting point for other NLP tasks.

A common approach to transfer learning in NLP is to train a language model (LM) on large amounts of unsupervised text (Howard and Ruder, 2018) and use it, with or without further fine-tuning, to solve other downstream tasks. Building on top of a LM has proven to be very successful, producing state-of-the-art (SOTA) results (Liu et al., 2019; Trinh and Le, 2018) on benchmark datasets like GLUE (Wang et al., 2019) or WSC273 (Levesque et al., 2011).

In this work, we first show that fine-tunning existing LMs on WSCR is a surprisingly robust method of improving the capabilities of the LM to tackle WSC273 and WNLI. Secondly, we introduce a method for generating large-scale WSC-like examples. We use this method to create an $11M$ dataset[1] from English Wikipedia[2], which we further use together with WSCR for fine-tuning the pre-trained BERT LM (Devlin et al., 2018). We achieve accuracies of 72.2% and 71.9% on WSC273 and WNLI, improving the previous best solutions by 8.5% and 6.8%, respectively.

[1]Available at http://tiny.cc/cxar6y
[2]https://dumps.wikimedia.org/enwiki/ dump id: enwiki-20181201

2 Background

This section contains a detailed description of the WSC and its relaxed form, the Definite Pronoun Resolution problem, as well as of the main LM used in our work, BERT (Devlin et al., 2018).

BERT. Our work uses the pre-trained Bidirectional Encoder Representations from Transformers (BERT) LM (Devlin et al., 2018) based on the transformer architecture (Vaswani et al., 2017). Due to its high performance on natural language understanding (NLU) benchmarks and the simplicity to adapt its objective function to our fine-tuning needs, we use BERT throughout this work.

BERT is originally trained on two tasks: masked token prediction, where the goal is to predict the missing tokens from the input sequence, and next sentence prediction, where the model is given two sequences and asked to predict whether the second sequence follows after the first one.

We focus on the first task to fine-tune BERT using WSC-like examples. We use masked token prediction on a set of sentences that follow the WSC structure, where we aim to determine which of the candidates is the correct replacement for the masked pronoun. We use the PyTorch implementation[3] of Devlin et al.'s (2018) pre-trained model, BERT-large.

Winograd Schema Challenge. Having introduced the goal of the Winograd Schema Challenge, we illustrate it with the following example:

The trophy didn't fit into the suitcase because ***it*** *was too [large/small].*

Question: What was too [large/small]?

Answer: the trophy / the suitcase

The pronoun "**it**" refers to a different noun, based on the word in the brackets. To correctly answer both versions, one must understand the meaning of the sentence and relation to the changed word. More specifically, a text must meet the following criteria to be considered for a Winograd Schema (Levesque et al., 2011):

1. Two parties must appear in the text.
2. A pronoun appears in the sentence and refers to one party. It would be grammatically correct if the pronoun referred to the other.
3. The question asks to determine what party the pronoun refers to.
4. A "special word" appears in the sentence. When switched to an "alternative word", the sentence remains grammatically intact but the referent of the pronoun changes.

Additionally, commonsense reasoning must be required to answer the question.

However, a detailed analysis by Trichelair et al. (2018) shows that not all WSC273 examples are equally difficult. They introduce two complexity measures (associativity and switchability) and, based on them, refine evaluation metrics for WSC273.

In *associative* examples, one of the parties is more commonly associated with the rest of the question than the other one. Such examples are seen as "easier" than the rest and represent 13.5% of WSC273. The remaining 86.5% of WSC273 is called *non-associative*.

47% examples are called "switchable", because the roles of the parties can be changed, and examples still make sense. A model is tested on the original, "unswitched" switchable subset and on the same subset with switched parties. The consistency between the two results is computed by comparing how often the model correctly changes the answer when the parties are switched.

Definite Pronoun Resolution. Since collecting examples that meet the criteria for WSC is hard, Rahman and Ng (2012) relax the criteria and construct the Definite Pronoun Resolution (DPR) dataset, following the structure of the WSC, but also accepting easier examples. The dataset, referred throughout the paper as WSCR, is split into a training set with 1322 examples and test set with 564 examples. We use them for fine-tuning the LMs and validation, respectively.

WNLI. One of the 9 GLUE benchmark tasks (Wang et al., 2019), WNLI is very similar to the WSC273 dataset but is phrased as an entailment problem instead. A WSC schema is given as a premise. The hypothesis is constructed by extracting the sentence part where the pronoun is, and replacing the pronoun with one candidate. The label is 1, if the candidate is the correct replacement, and 0, otherwise.

3 Related Work

There have been several attempts at solving WSC273. Previous work is based on Google queries for knowledge (Emami et al., 2018) (58%),

[3] https://github.com/huggingface/pytorch-pretrained-BERT

sequence ranking (Opitz and Frank, 2018) (63%), and using an ensemble of LMs (Trinh and Le, 2018) (63%).

A critical analysis (Trichelair et al., 2018) showed that the main reason for success when using an ensemble of LMs (Trinh and Le, 2018) was largely due to imperfections in WSC273, as discussed in Section 2.

The only dataset similar to WSC273 is an easier but larger (1886 examples) variation published by Rahman and Ng (2012) and earlier introduced as WSCR. The sequence ranking approach uses WSCR for training and attempts to generalize to WSC273. The gap in performance scores between WSCR and WSC273 (76% vs. 63%) implies that examples in WSC273 are much harder.

Another important NLU benchmark is GLUE (Wang et al., 2019), which gathers 9 tasks and is commonly used to evaluate LMs. The best score has seen a huge jump from 0.69 to over 0.82 in a single year. However, WNLI is a notoriously difficult task in GLUE and remains unsolved by the existing approaches. None of the models have beaten the majority baseline at 65.1, while human performance lies at 95.9 (Nangia and Bowman, 2018).

4 Our Approach

WSC Approach. We approach WSC by fine-tuning the BERT LM (Devlin et al., 2018) on the WSCR training set and further on a very large Winograd-like dataset that we introduce. Below, we present our fine-tuning objective function and the introduced dataset.

Given a training sentence $\mathbf{s}$, the pronoun to be resolved is masked out from the sentence, and the LM is used to predict the correct candidate. Let c_1 and c_2 be the two candidates. BERT for Masked Token Prediction is used to find $\mathbb{P}(c_1|\mathbf{s})$ and $\mathbb{P}(c_2|\mathbf{s})$. If a candidate consists of several tokens, the corresponding number of `[MASK]` tokens is used in the masked sentence. Then, $\log \mathbb{P}(c|\mathbf{s})$ is computed as the average of log-probabilities of each composing token. If c_1 is correct, and c_2 is not, the loss is:

$$L = -\log \mathbb{P}(c_1|\mathbf{s}) + \\ + \alpha \cdot \max(0, \log \mathbb{P}(c_2|\mathbf{s}) - \log \mathbb{P}(c_1|\mathbf{s}) + \beta) \quad (1)$$

where α and β are hyperparameters.

MaskedWiki Dataset. To get more data for fine-tuning, we automatically generate a large-scale collection of sentences similar to WSC. More specifically, our procedure searches a large text corpus for sentences that contain (at least) two occurrences of the same noun. We mask the second occurrence of this noun with the `[MASK]` token. Several possible replacements for the masked token are given, for each noun in the sentence different from the replaced noun. We thus obtain examples that are structurally similar to those in WSC, although we cannot ensure that they fulfill all the requirements (see Section 2). We partially address this challenge by using the BERT_WSCR LM, i.e., BERT fine-tuned on WSCR. Details on fine-tuning are given in Section 5. We rank the likelihoods of sentences obtained with the various possible replacements. We aim to only keep the "hard" examples, i.e., examples with the most likely alternative candidate, according to BERT_WSCR. The details of the filtering process are described in the supplementary material.

To generate such sentences, we choose the English Wikipedia as source text corpus, as it is a large-scale and grammatically correct collection of text with diverse information. We use the Stanford POS tagger (Manning et al., 2014) for finding nouns. After filtering, the generated dataset consists of $11,700,317$ examples.

WNLI Approach. For format consistency reasons, we transform WNLI examples from the premise–hypothesis format into the masked words format. The description of the transformation can be found in the supplementary material.

5 Evaluation

The training procedure differs from the training of BERT (Devlin et al., 2018) in a few points. The model is only trained with a single epoch of the MaskedWiki dataset, using batches of size 64 (distributed on 8 GPUs), learning rate of $3.0 \cdot 10^{-5}$, and hyperparameter values $\alpha = 5$ and $\beta = 0.1$ in the loss function (Eq. (1)). The values were selected manually from $\alpha \in \{2.5, 5, 10, 20\}$ and $\beta \in \{0.05, 0.1, 0.2, 0.4\}$ by comparing the training accuracy on a subset of size $2,000,000$.

Both BERT and BERT_Wiki are fine-tuned on the WSCR train dataset to create BERT_WSCR and BERT_Wiki_WSCR.

The WSCR test set was used as the validation set. The fine-tuning procedure was the same as the training procedure on MaskedWiki, except that (i) 50 epochs combined with early stopping were used, (ii) the hyperparameters α and β were se-

	WSC273	non-assoc.	assoc.	unswitched	switched	consist.	WNLI
BERT_Wiki	0.546	0.525	0.676	0.473	0.489	0.252	0.644
BERT_Wiki_WSCR	**0.722**	**0.716**	**0.757**	**0.748**	**0.725**	**0.611**	**0.719**
BERT	0.601	0.589	0.676	0.580	0.573	0.443	0.651
BERT_WSCR	**0.703**	**0.708**	**0.676**	**0.733**	**0.701**	**0.595**	**0.705**
BERT-base	0.564	0.551	0.649	0.527	0.565	0.443	0.630
BERT-base_WSCR	**0.656**	**0.636**	**0.784**	**0.687**	**0.649**	**0.489**	**0.651**
GPT	0.553	0.525	0.730	0.595	0.519	0.466	0.432
GPT_WSCR	**0.674**	**0.653**	**0.811**	**0.664**	**0.580**	**0.641**	0.432
BERT_Wiki_WSCR_no_pairs	0.604	0.606	0.595	0.611	0.611	0.420	–
BERT_Wiki_WSCR_pairs	**0.678**	**0.674**	**0.703**	**0.695**	**0.687**	**0.519**	–
LM ensemble	0.637	0.606	**0.838**	0.634	0.534	0.443	–
Knowledge Hunter	0.571	0.583	0.5	0.588	0.588	**0.901**	–

Table 1: Results on WSC273 and its subsets. Comparison between each language model and its WSCR-tuned model is given. For each column, the better result of the two is in bold. The best result in the column overall is underlined. Results for LM ensemble and Knowledge Hunter are taken from Trichelair et al. (2018). All models consistently improve their accuracy when fine-tuned on the WSCR dataset.

lected with grid search from the same sets, and (iii) the learning rate $2.0 \cdot 10^{-5}$ was used.

For comparison, experiments are also conducted on two other LMs, BERT-base (BERT with less parameters) and General Pre-trained Transformer (GPT) by Radford et al. (2018). The training on BERT-base was conducted in the same way as for other models. When using GPT, the probability of a word belonging to the sentence $\mathbb{P}(c|\mathbf{s})$ is computed as partial loss in the same way as by Trinh and Le (2018).

Due to WSC's "special word" property, examples come in pairs. A pair of examples only differs in a single word (but the correct answers are different). The model BERT_Wiki_WSCR_no_pairs is the BERT_Wiki model, fine-tuned on WSCR, where only a single example from each pair is retained. The size of WSCR is thus halved. The model BERT_Wiki_WSCR_pairs is obtained by fine-tuning BERT_Wiki on half of the WSCR dataset. This time, all examples in the subset come in pairs, just like in the unreduced WSCR dataset.

We evaluate all models on WSC273 and the WNLI test dataset, as well as the various partitions of WSC273, as described in Section 2. The results are reported in Table 1 and will be discussed next.

Discussion. We note that models that are fine-tuned on the WSCR dataset consistently outperform their non-fine-tuned counterparts. The BERT_Wiki_WSCR model outperforms other language models on 5 out of 7 sets that they are compared on. In comparison to the LM ensemble by Trinh and Le (2018), the accuracy is more consistent between associative and non-associative subsets and less affected by the switched parties. However, it remains fairly inconsistent, which is a general property of LMs.

The results of BERT_Wiki seem to indicate that this dataset is hurting BERT. However, when additionally fine-tuned to WSCR, the accuracy strongly and consistently improves. The results of BERT_Wiki_no_pairs and BERT_Wiki_pairs show that the existence of WSC-like pairs in the training data affects the performance of the trained model. MaskedWiki does not contain such pairs.

Summary and Outlook. This work achieves new SOTA results on the WSC and WNLI datasets by fine-tuning the BERT language model on the WSCR dataset and a newly introduced MaskedWiki dataset. The previous SOTA results on WSC and WNLI are improved by 8.5% and 6.8%, respectively. To our knowledge, this is the first model that beats the majority baseline on WNLI.

We show that by fine-tuning on WSC-like data, the language model's performance on WSC strongly improves. In future work, other uses of the MaskedWiki dataset and applications to different tasks will be investigated. Furthermore, to further improve the results on WSC273, the filtering procedure can be improved to generate harder WSC-like examples.

References

Ernest Davis, Leora Morgenstern, and Charles L Ortiz. 2017. The first winograd schema challenge at ijcai-

16. *AI Magazine*, 38(3):97–98.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *Computing Research Repository*, arXiv:1810.04805.

Ali Emami, Noelia De La Cruz, Adam Trischler, Kaheer Suleman, and Jackie Chi Kit Cheung. 2018. A knowledge hunting framework for common sense reasoning. *Computing Research Repository*, arXiv:1810.01375.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *Computing Research Repository*, arXiv:1803.05567.

Jeremy Howard and Sebastian Ruder. 2018. Fine-tuned language models for text classification. *Computing Research Repository*, arXiv:1801.06146.

Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of ACL*, pages 562–570. ACL.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd Schema Challenge. In *Proceedings of KR*. AAAI Press.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2011. The winograd schema challenge. *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 46.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *Computing Research Repository*, arXiv:1901.11504.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Leora Morgenstern, Ernest Davis, and Charles L. Ortiz. 2016. Planning, executing and evaluating the winograd schema challenge. *AI Magazine*.

Nikita Nangia and Samuel R. Bowman. 2018. A conservative human baseline estimate for glue: People still (mostly) beat machines.

Juri Opitz and Anette Frank. 2018. Addressing the winograd schema challenge as a sequence ranking task. In *Proceedings of the First International Workshop on Language Cognition and Computational Models*, pages 41–52. ACL.

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

Altaf Rahman and Vincent Ng. 2012. Resolving complex cases of definite pronouns: The winograd schema challenge. In *Proceedings of EMNLP*.

Paul Trichelair, Ali Emami, Jackie Chi Kit Cheung, Adam Trischler, Kaheer Suleman, and Fernando Diaz. 2018. On the evaluation of common-sense reasoning in natural language understanding. *Computing Research Repository*, arXiv:1811.01778.

T. H. Trinh and Q. V. Le. 2018. A Simple Method for Commonsense Reasoning. *Computing Research repository*, arXiv:1806.02847.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Computing Research Repository*, arXiv:1706.03762.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In the Proceedings of ICLR.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *Computing Research Repository*, arXiv:1609.08144.

Appendix

Dataset Filtering Procedure

We filter the dataset to reduce the number of "simple" examples, while capturing the "harder" ones. Let $\mathbf{s}$ denote a masked sentence, and let c_1, c_2 be the correct and incorrect candidate, respectively. For each example, $\log \mathbb{P}(c_1|\mathbf{s})$ and $\log \mathbb{P}(c_2|\mathbf{s})$ are computed using BERT_WSCR. We observe the value $v = \log \mathbb{P}(c_2|\mathbf{s}) - \log \mathbb{P}(c_1|\mathbf{s})$.

We manually observe 1000 randomly selected examples to determine how v relates to the quality of the example. More specifically, examples where one option is grammatically incorrect or is visibly a better choice than the other, are considered "easy", and the examples where reasoning or understanding is needed to solve them are considered "hard". A more detailed description of this criterion is included below.

We pick the examples with $-0.075 \leq v \leq 0.30$, where at least 90% of WordPiece tokens (Wu et al., 2016) represent whole words. The upper bound was necessary, because BERT_WSCR scored some incorrect examples as having high $\mathbb{P}(c_2|\mathbf{s})$, just because the correct solution was a really rare word, for example, a non-English surname with non-ASCII characters (hence, its predicted $\mathbb{P}(\mathbf{c}_1|\mathbf{s})$ was low). The boundary values were selected manually to retain hard examples and filter out the easy ones, based on manual inspection of the mentioned random subset. After filtering, the final dataset consists of $11,700,317$ out of a total of $129,852,279$ examples, i.e., we kept 9% of the initially generated dataset.

To determine the quality of the dataset, 200 random examples are manually categorized into 4 categories:

- Unsolvable: the masked word cannot be unambiguously selected with the given context. Example: *Mostly shot in Australia and South Africa, the film is based on the controversy regarding the allegedly racial attacks on Indian students in [MASK] between 2007 and 2010. [Australia/South Africa]*
- Hard: the answer is not trivial to figure out, but we do not require it to pass the Google test. Example: *The heavy grazing by the cattle resulted stoppage of regeneration of new grasses due to no seeding of seeds and trampling of new [MASK]. [grasses/seeds]*
- Easy: The alternative sentence is grammatically incorrect or is very visibly an inferior choice. Example: *"Stay on These Roads" achieved Platinum status in Brazil and Gold in the UK, Switzerland, the Netherlands and Germany and Double [MASK] in France. [Platinum Status/Switzerland]*
- Noise: The example is a result of a parsing error.

In the analyzed subset, 12% of examples were unsolvable, 53% were hard, 34% were easy, and 1% fell into the noise category.

Evaluation on WNLI

Models are additionally tested on the test partition of the WNLI dataset. To use the same evaluation approach as for the WSC273 dataset, the examples in WNLI have to be transformed from the premise–hypothesis format into the masked words format. Since each hypothesis is just a sub-string of the premise with the pronoun replaced for the candidate, finding the replaced pronoun and one candidate can be done by finding the hypothesis as a sub-string of the premise. All other nouns in the sentence are treated as alternative candidates. The Stanford POS-tagger (Manning et al., 2014) is used to find the nouns in the sentence. The probability for each candidate is computed to determine whether the candidate in the hypothesis is the best match. Only the test partition of the WNLI dataset is used, because it does not overlap with WSC273. We do not train or validate on the WNLI training and validation sets, because some of the examples share the premise. Indeed, when upper rephrasing of the examples is used, the training, validation, and test sets start to overlap.