

# A Surprisingly Robust Trick for the Winograd Schema Challenge

Vid Kocijan<sup>1</sup>, Ana-Maria Crețu<sup>2</sup>, Oana-Maria Camburu<sup>1,3</sup>, Yordan Yordanov<sup>1</sup>, Thomas Lukasiewicz<sup>1,3</sup>

<sup>1</sup>University of Oxford

<sup>2</sup>Imperial College London

<sup>3</sup>Alan Turing Institute, London

firstname.lastname@cs.ox.ac.uk, a.cretu@imperial.ac.uk

## 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 consistently and robustly 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 72.5% and 74.7% on WSC273 and WNLI, improving the previous state-of-the-art solutions by 8.8% and 9.6%, respectively. Furthermore, our fine-tuned models are also consistently more accurate 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-tuning existing LMs on WSCR is a robust method of improving the capabilities of the LM to tackle WSC273 and WNLI. This is surprising, because previous attempts to generalize from the WSCR dataset to WSC273 did not achieve a major improvement (Opitz and Frank, 2018). Secondly, we introduce a method for generating large-scale WSC-like examples. We use this method to create a 2.4M dataset from English Wikipedia<sup>1</sup>, which we further use together with WSCR for fine-tuning the pre-trained BERT LM (Devlin et al., 2018). The dataset will be made publicly available. We achieve accuracies of 72.5% and 74.7% on WSC273 and WNLI, improving the previous

<sup>1</sup>[https://dumps.wikimedia.org/enwiki/dump\\_id:enwiki-20181201](https://dumps.wikimedia.org/enwiki/dump_id:enwiki-20181201)

best solutions by 8.8% and 9.6%, respectively.

## 2 Background

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

**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.

**Winograd Schema Challenge.** Having introduced the goal of the Winograd Schema Challenge in Section 1, 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 its 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 correct, but the referent of the pronoun changes.

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

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% of the examples are “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 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. Six examples in the WSCR training set reappear in WSC273. We remove these examples from WSCR. We use the WSCR training and test sets for fine-tuning the LMs and for 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%),

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. We note that Opitz and Frank (2018) do not report removing the overlapping examples between WSCR and WSC273.

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 pre-trained 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  $s$ , the pronoun to be resolved is masked out from the sentence, and the LM is used to predict the correct candidate in the place of the masked pronoun. Let  $c_1$  and  $c_2$  be the two candidates. BERT for Masked Token Prediction is used to find  $\mathbb{P}(c_1|s)$  and  $\mathbb{P}(c_2|s)$ . If a candidate consists of several tokens, the corresponding number of [MASK] tokens is used in the masked sentence. Then,  $\log \mathbb{P}(c|s)$  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:

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

where  $\alpha$  and  $\beta$  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).

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. We obtain a dataset with approximately 130M examples. We downsample the dataset uniformly at random to obtain a dataset of manageable size. After downsampling, the dataset consists of 2.4M examples. All experiments are conducted with this downsampled dataset only.

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: *Palmer and Crenshaw both used Wilson 8802 putters , with [MASK] ’s receiving the moniker “ Little Ben ” due to his proficiency with it . [Palmer/Crenshaw]*
- **Hard:** the answer is not trivial to figure out. Example: *At the time of Plath ’s suicide , Assia was pregnant with Hughes ’s child , but she had an abortion soon after [MASK] ’s death . [Plath/Assia]*
- **Easy:** The alternative sentence is grammatically incorrect or is very visibly an inferior choice. Example: *The syllables are pronounced strongly by Gaga in syncopation while her vibrato complemented Bennett’s characteristic jazz vocals and swing . Olivier added , “ [MASK] ’s voice , when stripped of its bells and whistles, showcases a timelessness that lends itself well to the genre . ” [Gaga/syncopation]*
- **Noise:** The example is a result of a parsing error.

In the analyzed subset, 8.5% of examples were unsolvable, 45% were hard, 45.5% were easy, and 1% fell into the noise category.

**WNLI Approach.** Models are additionally tested on the test set of the WNLI dataset. To use the same evaluation approach as for the WSC273 dataset, we transform the examples in WNLI from the premise–hypothesis format into the masked words format. Since each hypothesis is just a substring 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 substring 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 set 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.

## 5 Evaluation

In this work, we use the PyTorch implementation<sup>2</sup> of Devlin et al.’s (2018) pre-trained model, BERT-large. To obtain BERT\_WIKI, we train on MaskedWiki starting from the pre-trained BERT. The training procedure differs from the training of BERT (Devlin et al., 2018) in a few points. The model is trained with a single epoch of the MaskedWiki dataset, using batches of size 64 (distributed on 8 GPUs), Adam optimizer, a learning rate of  $5.0 \cdot 10^{-6}$ , and hyperparameter values  $\alpha = 20$  and  $\beta = 0.2$  in the loss function (Eq. (1)). The values were selected from  $\alpha \in \{5, 10, 20\}$  and  $\beta \in \{0.1, 0.2, 0.4\}$  and learning rate from  $\{3 \cdot 10^{-5}, 1 \cdot 10^{-5}, 5 \cdot 10^{-6}, 3 \cdot 10^{-6}\}$  using grid search. To speed up the hyperparameter search, the training (for hyperparameter search only) is done on a randomly selected subset of size 100,000. The performance is then compared on the WSCR test set.

Both BERT and BERT\_WIKI are fine-tuned on the WSCR training dataset to create BERT\_WSCR

<sup>2</sup><https://github.com/huggingface/pytorch-pretrained-BERT>

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 30 epochs were used. The model was validated after every epoch, and the model with highest performance on the validation set was retained. The hyperparameters  $\alpha$  and  $\beta$  and learning rate were selected with grid search from the same sets as for MaskedWiki training.

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 the other models. When using GPT, the probability of a word belonging to the sentence  $\mathbb{P}(c|s)$  is computed as partial loss in the same way as by Trinh and Le (2018).

Due to WSCR’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 subsets of WSC273, as described in Section 2. The results are reported in Table 1 and will be discussed next.

**Discussion.** Firstly, 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 6 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.

Secondly, the results of BERT\_WIKI seem to indicate that this dataset alone does not help BERT. However, when additionally fine-tuned to WSCR, the accuracy consistently improves.

Finally, the results of BERT\_WIKI\_no\_pairs

	WSC273	non-assoc.	assoc.	unswitched	switched	consist.	WNLI
BERT_WIKI	0.619	0.597	0.757	0.573	0.603	0.389	0.712
BERT_WIKI_WSCR	<b><u>0.725</u></b>	<b><u>0.720</u></b>	<b><u>0.757</u></b>	<b><u>0.732</u></b>	<b><u>0.710</u></b>	<b><u>0.550</u></b>	<b><u>0.747</u></b>
BERT	0.619	0.602	0.730	0.595	0.573	0.458	0.658
BERT_WSCR	<b>0.714</b>	<b>0.699</b>	<b>0.811</b>	<b>0.695</b>	<b>0.702</b>	<b>0.550</b>	<b>0.719</b>
BERT-base	0.564	0.551	0.649	0.527	0.565	0.443	0.630
BERT-base_WSCR	<b>0.623</b>	<b>0.606</b>	<b>0.730</b>	<b>0.611</b>	<b>0.634</b>	<b>0.443</b>	<b>0.705</b>
GPT	0.553	0.525	0.730	0.595	0.519	0.466	–
GPT_WSCR	<b>0.674</b>	<b>0.653</b>	<b>0.811</b>	<b>0.664</b>	<b>0.580</b>	<b>0.641</b>	–
BERT_WIKI_WSCR_no_pairs	0.663	0.669	0.622	0.672	0.641	0.511	–
BERT_WIKI_WSCR_pairs	<b>0.703</b>	<b>0.695</b>	<b>0.757</b>	<b>0.718</b>	<b>0.710</b>	<b>0.565</b>	–
LM ensemble	0.637	0.606	<b><u>0.838</u></b>	0.634	0.534	0.443	–
Knowledge Hunter	0.571	0.583	0.5	0.588	0.588	<b><u>0.901</u></b>	–

Table 1: Results on WSC273 and its subsets. The 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 the 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.

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.

## 6 Summary and Outlook

This work achieves new SOTA results on the WSC273 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 WSC273 and WNLI are improved by 8.8% and 9.6%, 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 consistently improves. The consistent improvement of several language models indicates the robustness of this method. This is particularly surprising, because previous work (Opitz and Frank, 2018) implies that generalizing to WSC273 is hard.

In future work, other uses and the statistical significance of MaskedWiki’s impact and its applications to different tasks will be investigated. Furthermore, to further improve the results on WSC273, data-filtering procedures may be introduced to find harder WSC-like examples.

## Acknowledgments

This work was supported by the Alan Turing Institute under the UK EPSRC grant EP/N510129/1, by the EPSRC grant EP/R013667/1, by the EPSRC studentship OUCS/EPSRC-NPIF/VK/

1123106, and by an EPSRC Vacation Bursary. We also acknowledge the use of the EPSRC-funded Tier 2 facility JADE (EP/P020275/1).

## 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 J. Levesque, Ernest Davis, and Leora Morgenstern. 2011. The Winograd Schema Challenge. *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 46.
- Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd Schema Challenge. In *Proceedings of KR*. AAAI Press.
- 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 *Proceedings of ICLR*.