Does the Objective Matter? Comparing Training Objectives for Pronoun Resolution

Yordan Yordanov[1], Oana-Maria Camburu[1,2], Vid Kocijan[1], Thomas Lukasiewicz[1,2]
[1]University of Oxford, Oxford, UK
[2]Alan Turing Institute, London, UK
firstname.lastname@cs.ox.ac.uk

Abstract

Hard cases of pronoun resolution have been used as a long-standing benchmark for commonsense reasoning. In the recent literature, pre-trained language models have been used to obtain state-of-the-art results on pronoun resolution. Overall, four categories of training and evaluation objectives have been introduced. The variety of training datasets and pre-trained language models used in these works makes it unclear whether the choice of training objective is critical. In this work, we make a fair comparison of the performance and seed-wise stability of four models that represent the four categories of objectives. Our experiments show that the objective of sequence ranking performs the best in-domain, while the objective of semantic similarity between candidates and pronoun performs the best out-of-domain. We also observe a seed-wise instability of the model using sequence ranking, which is not the case when the other objectives are used.

1 Introduction

Hard cases of pronoun resolution have been a long-standing problem in natural language processing, which has served as a performance benchmark for the research community (Levesque et al., 2012; Wang et al., 2018, 2019a). For example, the WinoGrande dataset (Sakaguchi et al., 2019) consists of pronoun resolution schemas that are constructed so that resolving them requires background knowledge and commonsense reasoning. In WinoGrande, the pronoun is obscured by "___" to remove gender and number cues. The task is to find the correct candidate for "___" out of two given candidates. For example:

> *John moved the couch from the garage to the backyard to create space. The ___ is small.* Candidates: garage, backyard.

Recently, supervised learning on top of pre-trained language models has been established as the main approach for pronoun resolution (Kocijan et al., 2019b,a; Sakaguchi et al., 2019). Under this type of approach, we identify four categories of objectives commonly used for pronoun resolution:

1. comparing the language model probabilities for each candidate (Kocijan et al., 2019b,a; He et al., 2019),
2. using semantic similarity between the pronoun and the candidates (Wang et al., 2019b; He et al., 2019),
3. using sequence ranking among the possible substituted sentences (Opitz and Frank, 2018; Sakaguchi et al., 2019), and
4. selecting a candidate based on the attentions of the pronoun in a transformer model (Klein and Nabi, 2019).

We list one representative model from each category. For 1, Kocijan et al. (2019b) use the BERT masked language model (Devlin et al., 2018) to produce the probabilities of the pronoun to be replaced with each of the two candidates. For 2, the Unsupervised Deep Structured Semantic Model (UDSSM-I) (Wang et al., 2019b) uses contextualized word embeddings produced by a bidirectional recurrent neural network (BiRNN), and then compares the word embedding of each candidate with the word embedding of the pronoun. For 3, RoBERTa-WinoGrande (Sakaguchi et al., 2019) encodes a pair of sentences (one for each candidate substituted in the input) by using RoBERTa (Liu et al., 2019) to determine which substitution is the correct one. Finally, the zero-shot Maximum Attention Score (MAS) model (Klein and Nabi, 2019) selects a candidate based on how much the pronoun attends to each candidate internally in BERT.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 4963–4969,
November 16–20, 2020. ©2020 Association for Computational Linguistics

The problem with all these objectives is that they have not been introduced under the same circumstances. They use different language models and word embeddings (e.g., BERT, RoBERTa, or BiRNN), and have been trained on different data (e.g., DPR (Rahman and Ng, 2012), WinoGrande, or no additional data). Therefore, it is unclear whether the choice of the objective function is essential for pronoun resolution tasks. Moreover, the seed-wise stability and the expected performance of these models have usually not been reported. However, seed-wise instability and performance variation are well-known problems when fine-tuning transformer-based models (Liu et al., 2020; Dodge et al., 2020).

In this work, we compare the performance and seed-wise stability of the four categories of training objectives for pronoun resolution on equal grounds. To do this, for category 4, we adapt to training the zero-shot MAS model. For category 2, we also introduce Coreference Semantic Similarity (CSS), which is a simplification and modification of UDSSM-I for transformer encoders. We select WinoGrande as our training and development dataset due to its large size (40,938 examples) and generalizability to other pronoun resolution tasks (Sakaguchi et al., 2019). We also use for testing the following well-established datasets: the Winograd Schema Challenge dataset (WSC) (Levesque et al., 2012) and the Definite Pronoun Resolution dataset (DPR) (Rahman and Ng, 2012). We choose as language model RoBERTa (Liu et al., 2019), as it significantly outperforms BERT on WinoGrande, WSC, and DPR (Sakaguchi et al., 2019).

Finally, our evaluations are done under an unprecedentedly large number of seeds (20).

2 Models

This section presents the four training objectives and the models[1] that represent each of them.

All four models share the RoBERTa[2] contextualized word embeddings. RoBERTa has an identical transformer architecture to BERT (Devlin et al., 2018), with the only difference being the training procedure. Hence, RoBERTa is a masked language model that outputs the probability distribution for filling a gap in the text (denoted by a "<mask>" token). Additionally, RoBERTa is a text encoder, with one output for each token of the input sentence. Three of the models (2.1, 2.3, and 2.4) use a multi-layer perceptron (MLP) classification "head", which takes some part of the encoder as input.

All four models use binary cross-entropy loss with a pair of probabilities as input, and the following target labels: sentence correctness for 2.1 and candidate correctness for 2.2, 2.3, and 2.4.

2.1 WinoGrande Sequence Ranking

We refer to the RoBERTa-WinoGrande model introduced by Sakaguchi et al. (2019) as WG-SR, since it has a sequence ranking objective. This model predicts which sentence of a pair of substituted sentences is more plausible. Each of the pair of sentences in the input of WG-SR is split in two before the substituted candidate. For example,

> <s> The city councilmen refused the demonstrators a permit because </s> </s> ___ feared violence. </s>,

where "___" is filled with each of the two candidates: "the city councilmen" or "the demonstrators".

The WG-SR code[3] is based on the RobertaForMultipleChoice model (Wolf et al., 2019), restricted to binary choice. This model consists of the pre-trained RoBERTa encoder and an MLP head based on the <s> (first) token of RoBERTa's output. The MLP has one hidden layer with tanh activation, hidden size matching that of the encoder, and one-dimensional output. The pair of input sentences $(\mathrm{S}_1, \mathrm{S}_2)$ thus produces a pair of values, which are then passed through a softmax to obtain the two sentence probabilities $P(\mathrm{S}_1)$ and $P(\mathrm{S}_2)$.

2.2 Binary Word Prediction

We denote by Binary Word Prediction (BWP) the model suggested by Liu et al. (2019) in their code repository[4] as a modification of the model from Kocijan et al. (2019b). Instead of using margin loss, BWP uses binary cross-entropy loss. We select this modified version, because it is claimed to be more robust by its authors, and it also has two fewer hyperparameters.

For a given (unsubstituted) input sentence, the BWP model estimates which of the two candidates is more likely to fill the gap "___". The input format is like in the following example, where "___" is

[1] The code is publicly available at: https://github.com/YDYordanov/WS-training-objectives.

[2] *roberta-large* from (Wolf et al., 2019)

[3] https://github.com/allenai/winogrande

[4] https://github.com/pytorch/fairseq/tree/master/examples/roberta/wsc

replaced by the "<mask>" token, to serve for the masked language model:

> <s> The city councilmen refused the demonstrators a permit because <mask> feared violence. </s>

With such an input, the RoBERTa masked language model returns the log-probability predictions at the "<mask>" token over the vocabulary. Of those predictions, only the ones corresponding to the two word candidates c_1 and c_2 are selected by BWP: $\log P_{\text{vocab}}(c_1)$ and $\log P_{\text{vocab}}(c_2)$. Here, the log-probability of each candidate is defined by averaging the log-probabilities of its tokens. Then, softmax is computed with inputs $\log P_{\text{vocab}}(c_1)$ and $\log P_{\text{vocab}}(c_2)$, which is how we define the pair of probabilities: $(P(c_1), P(c_2)) := (P_{\text{vocab}}(c_1)/(P_{\text{vocab}}(c_1) + P_{\text{vocab}}(c_2)), P_{\text{vocab}}(c_2)/(P_{\text{vocab}}(c_1) + P_{\text{vocab}}(c_2)))$.

2.3 Coreference Semantic Similarity

We propose Coreference Semantic Similarity (CSS), a modification of the training objective of the Unsupervised Deep Structured Semantic Model (UDSSM-I) (Wang et al., 2019b). Like UDSSM-I, the CSS objective works by comparison in the word embedding space, such that the candidate that is more similar to the embedding of the pronoun is selected. Unlike UDSSM-I, the CSS objective is simpler, with no attention weights on the tokens of the candidates. It also uses a transformer encoder instead of a recurrent neural network, which enables it to take advantage of state-of-the-art pre-trained language models.

The input format for this model is the same as for BWP (2.2). This input is used by RoBERTa to produce contextualized word embeddings. For each candidate c, we define its contextualized word embedding $\text{emb}(c)$ by averaging the contextualized word embeddings of its tokens.

For classification, we compare the similarity scores of the embeddings of the <mask> token with each of the two candidates c_1 and c_2, i.e., we compare $\text{sim}(\text{emb}(c_1), \text{emb}(\text{<mask>}))$ and $\text{sim}(\text{emb}(c_2), \text{emb}(\text{<mask>}))$ and select the candidate with greater similarity.

For the similarity score function, we use *additive alignment* (Bahdanau et al., 2014), i.e., $\text{sim}(x, y) := v^\top \tanh(Wx + Uy)$, with the trainable parameters: vector v, and matrices W and U, with hidden size equal to that of RoBERTa and output size of one.

During training, $\text{sim}(\text{emb}(c_1), \text{emb}(\text{<mask>}))$ and $\text{sim}(\text{emb}(c_2), \text{emb}(\text{<mask>}))$ are fed to a binary softmax function to obtain $P(c_1)$ and $P(c_2)$.

2.4 Maximum Attention Score

The Maximum Attention Score (MAS) model was originally developed for zero-shot evaluation of transformer models on pronoun disambiguation (Klein and Nabi, 2019). It uses the attentions of all layers of a transformer model to produce a maximum attention score for each candidate that summarizes how much the pronoun attends to a candidate. The candidate that is most attended is selected. We adapt this objective to be trainable by replacing the summary of attentions with an MLP over the concatenated masked attention tensors, followed by a binary classifier.

The input of MAS is the same as for BWP (2.2). Then, similarly to Klein and Nabi (2019), we extract the two attention tensors A_{c_1} and A_{c_2} given by the multi-layer RoBERTa attentions of the "<mask>" token to each of the two candidates c_1 and c_2, respectively. For each candidate c, the attention tensor A_c is defined as the average of the attention tensors of all tokens that form c. The two corresponding max-masking tensors M_{c_1} and M_{c_2} are then derived as follows: for $i = 1, 2$ and for each multi-index j of the tensor A_{c_i}, we set $M_{c_i}(j) = 1$, if $A_{c_i}(j) \geq A_{c_{3-i}}(j)$, and $M_{c_i}(j) = 0$, otherwise. We obtain the two corresponding max-masked tensors by the element-wise products: $B_{c_1} = A_{c_1} \circ M_{c_1}$ and $B_{c_2} = A_{c_2} \circ M_{c_2}$.

Unlike Klein and Nabi (2019), we introduce an MLP on top of the concatenated tensor $B = [B_{c_1}, B_{c_2}]$ for binary classification. The MLP has two hidden layers, tanh activation, hidden size the same as its input, and two-dimensional output. It is followed by a binary softmax function to produce the two candidate probabilities $P(c_1)$ and $P(c_2)$.

3 Experiments

For all four models, we select the best hyperparameters via grid search using 3 seeds, and then train the models with the best hyperparameters on 20 additional seeds. For WinoGrande, we use WG-dev (1,267 examples) for selecting the hyperparameters, and WG-train-XL as our training dataset. Due to the submission limitation (maximum one per week) of the WinoGrande leaderboard,[5] we are unable to

[5] https://leaderboard.allenai.org/winogrande/submissions/public

Model	WG-dev	WSC	DPR
WG-SR	**78.2** (1.00)	89.2 (1.12)	92.2 (0.61)
BWP	76.3 (0.5)	89.6 (0.80)	91.8 (0.55)
CSS	77.4 (0.78)	**90.2** (0.90)	**92.7** (0.66)
MAS	76.6 (0.77)	89.0 (1.51)	92.3 (0.71)

Table 1: Seed-wise aggregated performance of models on WG-dev, WSC, and DPR. The number format is: average accuracy in %, and standard deviation (in parentheses). Out of the 20 seeds, only the converging ones are included. The best performance is marked in bold.

Model	Maximum	Average	Standard deviation	Number of converged
WG-SR	**80.0**	**76.8**	2.28	49 out of 96
BWP	77.6	75.4	1.45	54 out of 96
CSS	78.9	76.2	1.13	56 out of 96
MAS	77.7	74.5	2.50	69 out of 96

Table 2: Performance of all four models on WG-dev aggregated across all 96 hyperparameter combinations (including the three seeds). The numbers in the first three columns are: maximum accuracy in %, average accuracy in %, standard deviation. Only the converging models (with at least 60% accuracy) are reported, and their number is in the last column. The best performance is marked in bold.

report all 80 trained models on WG-test, and instead we report them on WG-dev. For additional verification, we include results over the hyperparameter space, where WG-dev is a true test set. We also report all models on the out-of-domain pronoun resolution datasets WSC (273 examples) and DPR (564 examples). The candidates provided in WSC were treated differently for the CSS and MAS models, as these models require precise candidate localization (see Appendix B).

For all four models, we do a grid search over the learning rate $\{5e-6, 1e-5, 3e-5, 5e-5\}$, the number of training epochs $\{3, 4, 5, 8\}$, and the batch-size $\{8, 16\}$, and we run each model with three different random seeds. This hyperparameter space is selected based on the union of the grid search by the original WG-SR work (Sakaguchi et al., 2019) and our observations on the other three models. The best hyperparameters (in Appendix A) are selected based on the maximum WG-dev accuracy across the three seeds.

For all experiments, we use linear learning rate decay with warm-up over 10% of the training data, and the AdamW optimizer (Wolf et al., 2019), for which we only alter the learning rate.

4 Results

Table 1 shows the final seed-wise results for all four objectives. We see that the semantic similarity objective (CSS) outperforms the other three objectives on out-of-domain testing, with 90.2% average accuracy on WSC and 92.7% average accuracy on DPR. On the other hand, the sentence ranking objective used by WG-SR clearly outperforms the other three objectives on in-domain testing, with 78.2% average accuracy on WG-dev. This is confirmed by the contents of Table 2, where we see that WG-SR has a better mean and max accuracy on WG-dev over the entire hyperparameter space compared to the other three models. For these cases, WG-dev is a true test set, since early stopping was not used, and all tested setups are reported; hence, WG-dev has not influenced the models reported in Table 2.

In order to verify the statistical significance of our main results, we used the t-test for similar variances and different sample sizes to compare the distributions of accuracy on the converging seeds. Comparing the accuracies of CSS and WG-SR on WG-dev, WSC, and DPR, respectively, we get the following two-tailed p-values: 0.008249, 0.003026, and 0.017441. All results are significant with $p < 0.05$.

We also observe that, even with the best hyperparameter combination, WG-SR exhibits seed-wise instability, as it fails to converge on 2 out of 20 seeds. This does not happen to the other three models. After considering 10 additional seeds, we obtained that WG-SR fails to converge on 10% of the seeds (3 out of 30).

Moreover, during the hyperparameter search, we observed that all models were prone to not converge for certain combinations of hyperparameters. The convergence threshold that we used was selected as having $\leq 60\%$ accuracy on WG-dev, and its value was selected based on the performance distribution of all models. We observed that all models either perform around 50% accuracy or 70% accuracy or more on WG-dev. 60% in this context is a good middle ground threshold. Table 2 shows that MAS converged most often; however, it also had the highest performance variation with a standard deviation of 2.5. Out of the four models, WG-SR converged least often, for only 49 out of all 96 hyperparameter combinations.

WG-SR likely performs better in-domain than CSS, MAS, and BWP, since those three use existing properties of RoBERTa (such as the possibility to compare contextualized embeddings, the attention structure of the model, and its pre-trained LM prediction head, respectively) for a task that they were not originally designed for (pronoun resolution). WG-SR, on the other hand, only uses the output of RoBERTa at the 0-th token, which is not pre-trained.

We identify two possible reasons why WG-SR performs worse than CSS on out-of-domain examples. The first reason is the one mentioned above, namely, not explicitly exploiting the listed properties of the pre-trained model would lead to a better fit on a specific dataset, but worse "general knowledge". This reason is not completely warranted, since WG-SR has similar out-of-domain performance to BWP and MAS. The second possible reason is that CSS uses an explicit candidate localization and candidate-pronoun matching (by comparing the embedding of the candidate and the pronoun), whereas in WG-SR these are achieved implicitly by feeding a pair of sentences to the model, one with the correct and one with the incorrect substitution. Again, this reason is not completely warranted, since MAS also uses explicit candidate localization and candidate-pronoun matching, but has a similar out-of-domain performance to WG-SR. Further investigation on the reasons why CSS outperforms WG-SR on the out-of-domain examples is left for future work.

5 Summary and Outlook

In this work, we categorized four existing objectives for pronoun resolution, and compared their performance and seed-wise stability on equal grounds. Our experiments showed that, on in-domain testing, the objective of sequence ranking based on the first token in RoBERTa outperforms the other three objectives, but can exhibit convergence problems. On out-of-domain testing, the objective of semantic similarity between the pronoun and each candidate outperforms the other three objectives.

Future work may investigate whether these results translate to other language models besides RoBERTa as well as other training datasets besides WinoGrande. Also, one could analyze the strengths and weaknesses of each objective, and evaluate other variations of these objectives.

Acknowledgments

This work was supported by a JP Morgan PhD Fellowship, the Alan Turing Institute under the EPSRC grant EP/N510129/1, the AXA Research Fund, the ESRC grant "Unlocking the Potential of AI for Law", the EPSRC studentship OUCS/EPSRC-NPIF/VK/1123106, and EU Horizon 2020 under the grant 952215. We also acknowledge the use of the EPSRC-funded Tier 2 facility JADE (EP/P020275/1).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *Computing Research Repository*, arXiv:1409.0473. Version 7.

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.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *Computing Research Repository*, arXiv:2002.06305.

Pengcheng He, Xiaodong Liu, Weizhu Chen, and Jianfeng Gao. 2019. A hybrid neural network model for commonsense reasoning. In *Proceedings of the 1st Workshop on Commonsense Inference in Natural Language Processing*, pages 13–21. Association for Computational Linguistics.

Tassilo Klein and Moin Nabi. 2019. Attention is (not) all you need for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Asso-*

ciation for Computational Linguistics, pages 4831–4836. Association for Computational Linguistics.

Vid Kocijan, Oana-Maria Camburu, Ana-Maria Cretu, Yordan Yordanov, Phil Blunsom, and Thomas Lukasiewicz. 2019a. WikiCREM: A large unsupervised corpus for coreference resolution. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4303–4312. Association for Computational Linguistics.

Vid Kocijan, Ana-Maria Cretu, Oana-Maria Camburu, Yordan Yordanov, and Thomas Lukasiewicz. 2019b. A surprisingly robust trick for the Winograd Schema Challenge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4837–4842. Association for Computational Linguistics.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd Schema Challenge. In *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning*, pages 552–561. AAAI Press.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. 2020. Understanding the difficulty of training transformers. *Computing Research Repository*, arXiv:2004.08249.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *Computing Research Repository*, arXiv:1907.11692.

Juri Opitz and Anette Frank. 2018. Addressing the Winograd Schema Challenge as a sequence ranking task. In *Proceedings of the 1st International Workshop on Language Cognition and Computational Models*, pages 41–52. Association for Computational Linguistics.

Altaf Rahman and Vincent Ng. 2012. Resolving complex cases of definite pronouns: The Winograd Schema Challenge. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 777–789. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. WinoGrande: An adversarial Winograd Schema Challenge at scale. *Computing Research Repository*, arXiv:1907.10641.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *Computing Research Repository*, arXiv:1905.00537. Version 3.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355. Association for Computational Linguistics.

Shuohang Wang, Sheng Zhang, Yelong Shen, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, and Jing Jiang. 2019b. Unsupervised deep structured semantic models for commonsense reasoning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 882–891. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. *Computing Research Repository*, arXiv:1910.03771. Version 4.

Model	epochs	batch size	learn. rate
WG-SR	5	16	1e-5
BWP	8	16	1e-5
CSS	8	16	1e-5
MAS	8	8	1e-5

Table 3: The best hyperparameters for every model.

A Best Hyperparameters

See Table 3 for the best hyperparameters for each model.

B WSC Preprocessing

When evaluating the CSS and the MAS model on the WSC dataset, we noticed a problem with the dataset, which interfered with locating the candidates in the text. The problem is that, in some WSC examples, the given candidate options do not match word-by-word the candidates as they appear in the text. For example,

> *Madonna fired her trainer because ___ couldn't stand her boyfriend.*
>
> Candidates: Madonna, The trainer.

In this example, we resolve this problem by manually replacing the candidate option "the trainer" with "her trainer", to match exactly the candidate as it appears in the text. By following this procedure, we manually modified all 88 problematic examples in WSC (out of 273 examples in total). Note that this problem does not exist for WinoGrande and DPR. Furthermore, in real-world applications, such a problem does not exist, since the candidates are not provided and have to be extracted automatically from the text. Detected candidates thus match the spans in the text.

We use this modified version of WSC only for the CSS and MAS models, because they require precise candidate localization. For WG-SR and BWP, we use the unmodified WSC version. The edited dataset can be found in the code repository[6].

[6] https://github.com/YDYordanov/WS-training-objectives