Exploiting Language Model Prompts Using Similarity Measures: A Case Study on the Word-in-Context Task

Mohsen Tabasi[1], **Kiamehr Rezaee**[2] and **Mohammad Taher Pilehvar**[3]

[1]Department of CE, Iran University of Science and Technology, Tehran, Iran
[2*]Cardiff NLP, School of Computer Science and Informatics, Cardiff University, UK
[3]Tehran Institute for Advanced Studies, Khatam University, Tehran, Iran
s_tabasi@comp.iust.ac.ir, rezaee.k@cardiff.ac.uk
mp792@cam.ac.uk

Abstract

As a recent development in few-shot learning, prompt-based techniques have demonstrated promising potential in a variety of natural language processing tasks. However, despite proving competitive on most tasks in the GLUE and SuperGLUE benchmarks, existing prompt-based techniques fail on the semantic distinction task of the Word-in-Context (WiC) dataset. Specifically, none of the existing few-shot approaches (including the in-context learning of GPT-3) can attain a performance that is meaningfully different from the random baseline. Trying to fill this gap, we propose a new prompting technique, based on similarity metrics, which boosts few-shot performance to the level of fully supervised methods. Our simple adaptation shows that the failure of existing prompt-based techniques in semantic distinction is due to their improper configuration, rather than lack of relevant knowledge in the representations. We also show that this approach can be effectively extended to other downstream tasks for which a single prompt is sufficient.[†]

1 Introduction

Recently, there has been a resurgence of interest in few-shot learning, especially after the introduction of GPT-3 (Brown et al., 2020). The current dominant few-shot approach is the so-called prompt-based learning which involves a simple reformulation of the target task as a cloze-style (Taylor, 1953) fill-in-the-blank objective. The core idea is to extract knowledge by asking the right question from the pre-trained language model (PLM) using a task-specific prompting template which directs the PLM to generate a textual output corresponding to a target class. This paradigm has proven its effectiveness in the few-shot setting, even for relatively smaller models, such as BERT (Devlin et al., 2019) and RoBERTA (Liu et al., 2019), when combined with ensembling and fine-tuning (Schick and Schütze, 2021a). From the practical point of view, prompt-based learning is particularly well-suited for massive models, such as GPT-3, since it does not involve parameter tuning.

Prompt-based techniques have shown impressive performance in the few-shot setting, especially when compared to standard fine-tuning on datasets of hundreds of data points (Le Scao and Rush, 2021). However, surprisingly, the Word-in-Context task (Pilehvar and Camacho-Collados, 2019) –one of the tasks in the SuperGLUE benchmark (Wang et al., 2019)– is one exception on which these methods fail to stay on par with their fine-tuned counterparts.[‡] While a simple fine-tuned BERT-base model achieves around 69% accuracy on this task (Wang et al., 2019), GPT-3, with more than 100 times the number of parameters, performs no better than a random baseline by employing a prompt-based approach (Brown et al., 2020). The same pattern of failure is also observed in the more recent prompt based attempts (Liu et al., 2021; Schick and Schütze, 2021a).

The natural question that arises here is if the failure of few-shot techniques on WiC is due to lack of relevant encoded knowledge in PLMs or the inefficiency of the employed prompt-based methods. Two issues could be responsible for the latter case: (1) improper prompt, or (2) inefficient utilization of PLM's response. To address the first issue, there have been proposals to automatically find a suitable prompt template using a search in the discrete token space (Shin et al., 2020) or in the continuous embedding space (Liu et al., 2021). However, none of these have shown success on the WiC task.

In this work we investigate the latter issue by

[*]Work done as a Master's student at IUST.

[†]The code is freely available at https://github.com/tabasy/similarity_prompting

[‡]Given an ambiguous target word in two different contexts, the task in WiC is defined as a simple binary classification problem to identify if the triggered meaning of the target word differs in the two contexts or not.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 325 - 332

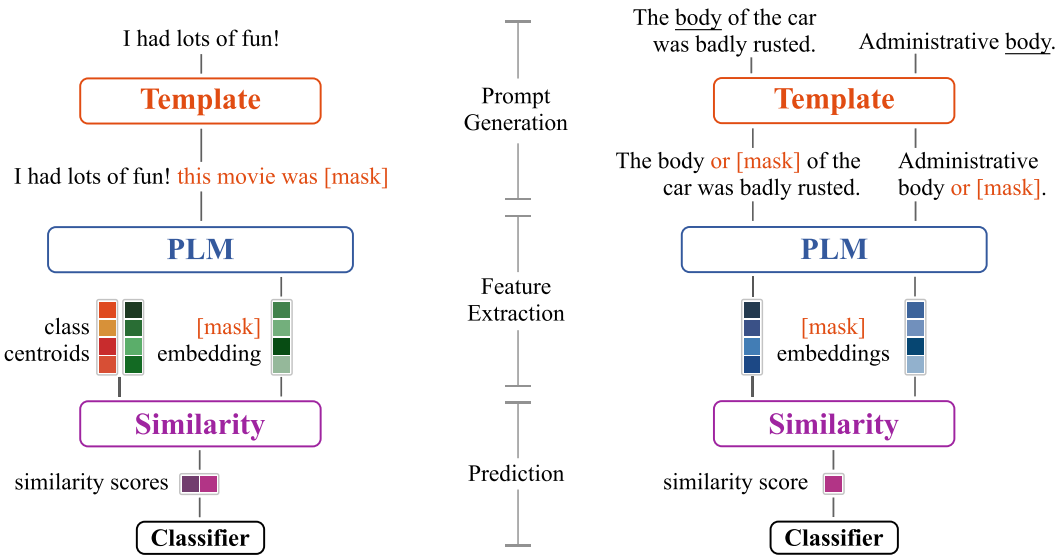

Figure 1: An illustration of the similarity-based method applied to sentiment analysis (left) and WiC (right).

introducing a new configuration for prompting. Given the comparison-based nature of WiC, we hypothesize that conventional prompting methods fall short since they only utilize a single prompt response. Hence, instead of relying on a single response, we make use of the similarity of PLM's response to the combination of a pair of prompts. The experimental results on the WiC dataset shows that, with only 16 instances per class, our proposed prompt-based technique can achieve comparable results to the fine-tuned models (with access to full training data of 2700+ instances per class). Moreover, we show that with few adjustments, this simple approach can be effectively used for other downstream tasks.

2 Methodology

Fine-tuning on a specific task can potentially update PLMs on what the task is and how to solve it. Assuming that PLMs know how to solve some tasks (to some extent), prompt-based learning focuses on the former, i.e., teaching the model what the task is, without needing to resort to large amounts of data or additional parameters. The common approach in prompt-based learning is to reformulate the task as a cloze-style question. For instance, to ask about the sentiment of a movie review, one can augment the review with a cloze question like "this movie was ——.". Existing methods often pick a set of one or few word predictions as a representative for each class, utilizing the language model's response in a sub-optimal manner. We propose a similarity-based method that not only better exploits the response, but also allows using multiple prompts which paves the way for comparison-based tasks, such as WiC. In what follows in this section, we describe our similarity-based prompting approach which we will refer to as **SP** (Similarity Prompting).

As shown in Figure 1, SP consists of three main steps: (1) prompt generation, (2) feature extraction, and (3) prediction. Given a task-specific input consisting of one or more text sequences, we first use a template function to generate a prompt—a sequence of tokens containing one [MASK] token—per input sequence. For instance, in sentiment analysis, for the movie review "Just give it a chance.", a valid template function would generate as output prompt: "Just give it a chance. this movie was ——.". The next step is feature extraction from a PLM. This is done by giving the generated prompts to the PLM as input and obtaining its contextualized embedding at the MASK index.

The third step is where SP differs from existing prompt-based approaches. Here, we first obtain class-specific centroids by taking the average of the MASK embeddings of our few training examples. To classify a new sample at inference time, a simple approach would be to employ a nearest centroid classifier. However, this assumes the variance of different classes to be equal in the embedding space. To alleviate the problem, we perform a class centroid-based dimension reduction (i.e. by taking

the similarity to each centroid as a feature), and train a simple linear classifier. This linear model is then used at inference time to evaluate SP on test set.

2.1 Similarity Prompting for WiC

The surprising failure of existing prompt-based techniques on the Word-in-Context task (Pilehvar and Camacho-Collados, 2019, WiC), motivated us to focus on filling this gap. Given an ambiguous target word in two different contexts, the task in WiC is defined as a simple binary classification problem to identify if the triggered meaning of the target word differs in the two contexts or not.

Previous work has fallen short of designing a single prompt template which make the PLM answer about the target word having the same meaning or not (e.g., with "yes" or "no"). Therefore, we ask PLM about the triggered meaning of the target word, separately for each context, and leave the comparison to similarity measures. Having an input sentence and the target word index, we insert "or ——" after the target word, where "——" indicates the MASK token. In the first step of SP, we apply this template function to both input sentences which generates a pair of prompts. Next the prompts are separately fed to PLM, resulting in a pair of mask embeddings as PLM's response. Finally, our classification step reduces to that of directly comparing our pair of embedding vectors using a similarity function, to produce a single similarity score for each instance. We then train the same linear model as before on the similarity scores of the training set examples to find the best discriminating threshold.

Similarity Measures. We opted for two similarity metrics: cosine similarity and Spearman's rank correlation. The latter is a rank-based comparison measure which is insensitive to the absolute values of individual dimensions (rather checks for their relative rankings).

3 Experiments

3.1 Comparison Systems

We compare our results on WiC with three other methods, all of which use 32 examples for their training. PET (Schick and Schütze, 2021b) prefers ALBERT-xxlarge-v2 (Lan et al., 2019) over RoBERTa (with an average gain of 8 points on a subset of SuperGLUE tasks) and fine-tunes it with manually engineered cloze-style prompts. P-tuning (Liu et al., 2021) uses the same PLM as PET, but optimizes a continuous prompt instead of tuning PLM parameters. GPT3 (Brown et al., 2020) is different in that it employs the so-called in-context learning which involves no parameter tuning.

3.2 Tasks

In addition to WiC, we also carried out experiments on two more tasks. The goal of this additional experiment is twofold: first, to show the applicability of SP to other settings, including tasks with single input sequence; and second, to evaluate if SP is effective when using prompt templates from other techniques, including those optimized for specific tasks. For this experiment, we compare against AutoPrompt (Shin et al., 2020). The approach makes use of full training set to optimize discrete prompts for each specific target task. Following AutoPrompt, we report results for the following two task:

SST. Stanford Sentiment Treebank (Socher et al., 2013) contains fine-grained sentiment labeled parse trees of sentences from movie reviews. Systems are evaluated either on a five-way fine-grained or binary classification task. We follow the latter (SST-2) in our experiments. For this task we used the automatically-generated template of AutoPrompt, along with the following manual template: $T(sent) = sent$ + " this movie was ——.", where $sent$ is the input sentence and "+" is concatenation operator. This is the same manual prompt used in AutoPrompt.

SICK. Sentences Involving Compositional Knowledge (Marelli et al., 2014) is a collection of sentence pairs annotated with their entailment relationship as well as a quantified measurement of their semantic similarity. In our experiments, we only use the former annotations (SICK-E) to compare our results with AutoPrompt, which only reports results for its optimized prompt. Thus we define our own manual template function as: $T(pre, hyp) = pre$ + "? Answer: ——, " + hyp, where pre is the premise and hyp is the hypothesis of an input example.

3.3 Setup

To train our models, we only used 16 examples per class. As for PLM, we opted for RoBERTA-large to be able to benchmark our results against AutoPrompt's (Shin et al., 2020). Our experiments are

Method	WiC	
	dev	test
Random Baseline	50.0	50.0
Fine-tuned RoBERTa-Large	-	69.9
GPT3 few-shot (Brown et al., 2020)	55.3	49.4
PET (Schick and Schütze, 2021b)	52.4	50.7
P-tuning (Liu et al., 2021)	56.3	-
Similarity Prompting - Cosine	60.3±0.4	63.6±0.5
Similarity Prompting - Spearman	**69.4±1.4**	**70.2±1.3**

Table 1: Accuracy percentage scores for Word-in-Context task. SP models are based on RoBERTa-Large.

Method	SST-2	SICK-E	
		Standard	Balanced
Majority baseline	50.0	56.7	33.3
Fine-tuned BERT	93.5	86.7	84.0
Manual Prompt			
AutoPrompt	85.2	-	-
SP-Cosine	89.1±2.1	77.3±1.5	79.8±0.8
SP-Spearman	89.2±1.8	76.6±2.3	79.0±1.0
Auto-generated Prompt			
AutoPrompt	91.4	65.0	69.3
SP-Cosine	90.7±2.3	62.1±1.0	63.2±1.9
SP-Spearman	91.8±1.5	61.6±0.7	62.2±1.6

Table 2: Test set accuracy on SST-2 and SICK-E tasks. SP and AutoPrompt (Shin et al., 2020) methods are based on RoBERTa-Large.

repeated 5 times using different randomly sampled training examples. For each experiment, we report the average performance along with the standard deviation.

3.4 Results

Given that our experiments are mainly focused on the WiC dataset, we first report our results on this benchmark, and then provide additional results for the other two tasks.

3.4.1 WiC

Table 1 summarizes the results on WiC with RoBERTa-Large as SP's PLM. The performance of SP in the few-shot setting is in the same ballpark as supervised fine-tuning (with nearly 170 times the data, i.e., 2,714 instances per class). This observation suggests that PLMs already encode a certain amount of task-related knowledge and the supervised fine-tuning mainly updates their task description (i.e., what the task is, not how to solve it). Therefore, using limited examples in the few-shot setting they are able to reach their maximum fine-tuning potential on WiC. We report SP's performance on WiC for other PLMs in the Appendix which shows our method/observation does not depend on a specific PLM. We also include some detailed examples of how SP works for WiC in the Appendix.

3.4.2 SICK and SST-2

The results on SST-2 and SICK-E are shown in Table 2. We compare SP with AutoPrompt which searches for the best template for each task. For SST-2, we observe that SP can exploit a manual prompt template significantly better than AutoPrompt, while being competitive using the best template optimized by AutoPrompt (auto-generated). This suggests that it is possible to gain significant improvement by simply exploiting a non-optimized manual prompt template.

To compare our results with AutoPrompt on the SICK-E task, we report accuracy score of SP for the standard test set (with neutral majority) and its balanced variant. SP retains an acceptable level of performance, particularly with the manual prompt, but lags behind with the auto-generated prompt. We note that the goal of this experiment was to showcase that our simple adaptation is also applicable to scenarios other than the setting of WiC. In fact, one could argue that the auto-generated prompt of AutoPrompt is sub-optimal for our model, which results in dropped performance on the SICK-E dataset.

3.5 Similarity Measures Comparison

Notably, the Spearman correlation score, which is less commonly used for comparing embeddings, outperforms the cosine similarity on WiC by a large margin while maintaining the same level of performance on other tasks. This superiority can be explained by the assumption that cosine similarity is more susceptible to variations in the dominant dimensions. To evaluate this hypothesis, we performed an experiment in which the most dominant dimension was set to zero for all the embeddings (the dominant dimension is identical across all vectors). The results approve the assumption: pruned cosine similarity gains around 10% absolute performance boost on WiC, filling the gap to Spearman correlation. However, the gain in the other two tasks is negligible.

The difference in the gain across tasks can be explained by the difference in their underlying nature.

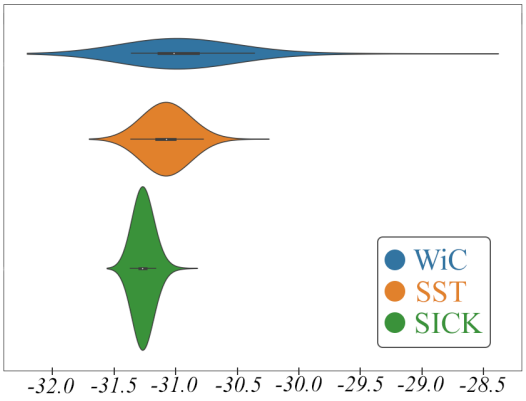

Figure 2: The distribution of values for the most dominant dimension of the MASK embedding for 1200 samples for the three tasks.

In WiC, the MASK embeddings can potentially refer to any word, varying from sample to sample. However, in SST and SICK the MASK template embedding is more restricted, often representing a closely related word to one of the class centroid embeddings (e.g., in SST the MASK embedding almost always represents a positive or negative adjective). This results in a higher spread on the most dominant dimension in the case of WiC. It is known that the most dominant dimensions in PLMs often encode irrelevant information, such as word frequency (Gao et al., 2019), therefore hampering performance for sensitive metrics such as cosine similarity. To verify our hypothesis, we ran an experiment using 1200 sample MASK embeddings for each of our three tasks. Figure 2 illustrates the distribution of values for the most dominant dimension. The ratio of variance is 6.5 times for WiC compared to SST and 27.3 times compared to SICK. This further supports the sensitivity of cosine similarity for WiC to the noisy variations along the most dominant dimension compared to the other two tasks.

4 Conclusion

We proposed an adaptation of prompt-based learning which addresses the common failure of existing techniques on the WiC dataset. In this work we showed that similarity based approach to prompt-based learning is capable of achieving comparable results to purely fine-tuning based methods on Word-in-Context task, in which previous few-shot attempts have failed. We also showed that Spearman's ranking correlation is a more robust choice of similarity measure compared to cosine similarity in this setting. We hope that our positive results inspire other prompting strategies to better exploit the encoded knowledge in PLMs. As future work, one interesting direction could be to perform further analysis on the behaviour of Spearman's correlation compared to cosine similarity anywhere it is applicable as a similarity measure.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. 2019. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Teven Le Scao and Alexander Rush. 2021. How many data points is a prompt worth? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2627–2636, Online. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too.

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.

M. Marelli, S. Menini, Marco Baroni, L. Bentivogli, R. Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *LREC*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. 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 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Wilson L. Taylor. 1953. "Cloze Procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *Advances in Neural Information Processing Systems*, 32.

A Experiments with other PLMs

This appendix contains more details on WiC experiments. Table 3 shows full test set results of SP for different PLMs and similarity measures to compare the performance of SP in different scenarios. Since our cloze-style prompt template is not applicable to GPT2, we use a different template for it: $sentence$ + $targetword$ + " means ——". The results in Table 3 generally confirm the effectiveness of SP with different PLMs. Notably, this observation is in line with our previous experiments that in general Spearman has superior performance over Cosine similarity.

Base model	**Cosine**	**Spearman**
RoBERTa-Large	63.6	70.2
BERT-Large-Cased	69.4	69.0
RoBERTa-Base	63.8	68.7
BERT-Base-Cased	64.8	67.1
GPT2-Large	56.4	63.3
GPT2-Base	62.3	62.6

Table 3: Test set accuracy of SP on WiC task, based on different PLMs (both Masked language model and Causal language models) and similarity metrics.

B Qualitative Analysis

We include some examples of how SP works on WiC in Table 4 for qualitative analysis. The examples are those from WiC dev set which had negative labels. We did not include the positive examples, since the observation that the same words with the same senses are treated similarly, might not provide a useful insight. The table presents our generated prompts, top-5 most probable words predicted by RoBERTa-Large for each prompt and the final prediction of SP. The top three examples are correctly predicted as negative with high confidence (high similarity score), while the bottom three are predicted positive again with high confidence. The most probable predicted words for the top three examples indicate that the PLM has spotted the correct senses in both contexts. For the bottom three where the model fails, we can observe that the target words have very similar or close senses, making them really hard to distinguish.

Prompt1 (Top-5 words)	**Prompt2 (Top-5 words)**	**Prediction**	**Ground Truth**
The drawing or —— of water from the well. (use, extraction, taking, pumping, consumption)	He did complicated pen-and-ink drawings or —— like medieval miniatures. (paintings, sculptures, something, more, looked)	Not matched	Not matched
The body or —— of the car was badly rusted. (trunk, roof, chassis, frame, grill)	Administrative body or ——. (agency, institution, government, commission, equivalent)	Not matched	Not matched
The main body of the sound or —— ran parallel to the coast. (river, bay, sea, ocean, channel)	He strained to hear the faint sounds or ——. (voices, footsteps, whispers, conversations, cries)	Not matched	Not matched
He could not conceal his hostility or ——. (anger, disgust, irritation, contempt, frustration)	He could no longer contain his hostility or ——. (anger, rage, frustration, aggression, disgust)	Matched	Not matched
There was a blockage or —— in the sewer, so we called out the plumber. (something, leak, obstruction, defect, overflow)	We had to call a plumber to clear out the blockage or —— in the drainpipe. (debris, obstruction, water, leak, crack)	Matched	Not matched
The senator received severe criticism or —— from his opponent. (threats, ridicule, mockery, attacks, threat)	The politician received a lot of public criticism or —— for his controversial stance on the issue. (backlash, ridicule, mockery, condemnation, criticism)	Matched	Not matched

Table 4: Detailed examples of how SP works on WiC.