Prompted Language Modeling for solving NLI

Wenkai Chen
w.chen5@students.uu.nl

Admitos Passadakis
r.a.passadakis@students.uu.nl

Abstract

LMs have demonstrated impressive performance on various NLP tasks. However, traditional fine-tune training procedure needs large amount of task-specific corpus as well as computational resources. In this paper, we present a zero-shot method to transfer the NLI task into a MLM prediction task using prompts. Initially, we ask the model to predict entailment and contradiction labels based on prompts. Subsequently, we integrate results from multiple prompts using a decision tree to refine the prediction. Our method demonstrates improvements over baselines on three popular NLI datasets. Additionally, we conduct further analysis to investigate the impact of different prompts and input lengths on the performance of the models. We find that guided prompts enhance the models' performance, while shorter input sequences have positive effect on their accuracy[1].

1 Introduction

Pre-trained transformer based language models (LMs), and in particular bidirectional masked LMs of the BERT family (Devlin et al., 2018; Liu et al., 2019a; Joshi et al., 2019; He et al., 2020), have revolutionized the landscape of Natural Language Processing (NLP). Models such as BERT, RoBERTa and DeBERTa are quite strong on different kind of tasks one of which is Mask Language Modelling (MLM). The latter, is an unsupervised learning technique which enables LMs to grasp contextual nuances and syntactic structures of textual data.

In this paper we explore how and if we can exploit MLM from pre-trained models for solving Natural Language Inference (NLI) tasks. NLI is a sub-category of Natural Language Understanding (NLU), which aims to decipher the complex interplay of meaning between 2 sentences which we call them premise and hypothesis. Leveraging pre-trained LMs specialized for MLM such as BERT and RoBERTa and BART (Lewis et al., 2019), we embark on a journey to predict NLI labels – entailment, neutral, and contradiction on three different datasets: Sentences Involving Compositional Knowledge - SICK (Marelli et al., 2014), Stanford Natural Language Inference - SNLI (Bowman et al., 2015) and MultiNLI - MNLI (Williams et al., 2017). For the sake of an exploratory and comprehensive research on some of our experiments we also deploy GPT2 (Radford et al., 2019) a model known for its ability to generate text autoregressively and not so much for MLM.

Rather than traditional supervised training, our methodology hinges on the innovative approach of prompting (Brown et al., 2020b; Madaan et al., 2022) whilst our models remain totally untrained. We employ several templated prompts like: "If 'Premise' is true then 'Hypothesis' is [MASK]", expecting the models to catch the logical relationships and prompt-specific semantic entailments. This approach capitalizes on the pre-trained contextual embeddings generated through MLM, providing the models with a solid comprehension of linguistic contexts, being at the same time extremely computationally efficient due to absence of fine-tuning.

Instead of training the heavy, in terms of parameters, language models, we demonstrate an alternative method where we guide our models towards classification which is based on the probability distribution given as an output over their respective vocabulary. In such way, we construct an aggregation of labelling votes made by the models. On top of that a standard classification method (Decision Trees) is used as a head classifier. We also include two more explorative steps. The first, exploits tokens which are believed to play crucial role in the final decisions of the models in order to boost their accuracy and in the second we investigate if the length of the templated prompts influence the efficacy of the language models in this task. Therefore, we hope the findings of our exploration to shed new light on the usability of prompted MLM for capturing linguistic nuances for NLI tasks.

2 Related Work

2.1 Few-shot Learning

Traditionally, LMs require fine-tuning on large corpus to excel in downstream NLP tasks (e.g. sen-

[1] A link to the code can be found by clicking HERE.

timent classification). However, with the development in NLP field, there has been a gradual reduction in reliance on large amounts of labeled data for downstream tasks. Recently published papers (Brown et al., 2020a; Petroni et al., 2019) have highlighted that models can achieve relatively good performance in a few-shot (zero-shot) setting. In this paper, we aim to address the Natural Language Inference (NLI) task in a zero-shot setting.

2.2 Prompt-based learning

Pre-trained models often perform bad in downstream tasks because they haven't seen to the specific task format. For example, it is challenging for a language model to generate a classification result in a multiple-choice question answering task without any task-specific data. To address this issue, (Schick and Schütze 2020) proposed a strategy to reformulate the original task inputs as close-style phrases. This approach transforms the original task into a Masked Language Modeling (MLM) prediction task, which aligns with the exact task that the pre-trained language model is trained on. In our work, we adopt a similar approach to transfer the NLI task into an MLM task by designing corresponding prompts.

3 Methods

It is plausible that NLI can be considered a particularly demanding task since it combines language understanding, logic or even general knowledge. In order to deal with this task we tried to exploit the idea of MLM of language models in a unsupervised manner that would make them understand the the nature of the problem in the best possible way. Here, we proposed a zero-shot prompting based method to solve NLI tasks.

3.1 Prompt Creation and Prediction

Inspired by prior research utilizing prompts or patterns (Schick and Schütze, 2021), we designed a bunch of different templates, denoted as $\mathbb{T} = \{T_1, T_2, ..., T_n\}$. For every input pair x, with premise and hypothesis $x = (p, h)$, we use the templates to reformat them into one single sentence, denoted as $T_i(x)$. For example, with an input pair $x =$ (two dogs are wrestling, there is no dog wrestling), we can form a sentence:

$T_1(x)$: Given that the p is *true*, then the h will be [MASK]

Given the provided template, the LM is required to do a Masked Language Modeling (MLM) task, where its objective is to generate the probability distribution for the [MASK] token. We aggregate the probabilities of a set of words, denoted as $Z = \{z_1, z_2, ..., z_n\}$, from the model's output. Here, each z_i represents a word[2] that corresponds to a specific label l_i in the set of labels $L = \{l_1, l_2, ..., l_n\}$, where n denotes both the total number of words in the set and the total number of labels. In NLI task, the value of n commonly equals 3, representing three distinct relationships: **entailment**, **contradiction**, and **neutral**. So far, the NLI task is now transferred to a MLM task, and it can be defined as follows:

$$l_{T_i} = \arg\max_{z_j \in Z} logP(z_j|T_i(x)) \tag{1}$$

Here, $Z = \{z_1, z_2, z_3\}$, T_i denotes i^{th} template and l_{T_i} is the LM's prediction for T_i. The target is to find the word z_j that maximizes the log probability of the mask token given the template.

To make the whole sentence sounds natural, we assume the simplest case where the word z_1="true" to signify entailment and z_2="false" to denote contradiction. However, for the neutral label, identifying a single specific word (e.g. "unknown") that seamlessly represents this category without compromising sentence naturalness is difficult. This presumption is also supported by the model's performance in the neutral label (see sub-section 5.2). We introduce a straightforward solution to address this problem. In the initial prediction step, we exclude the neutral label, getting a pseudo label l'_{T_i} which can only represent entailment and contradiction and then, we aim to map l'_{T_i} back to l_{T_i}.

3.2 Mapping Predictions

We assume that if the golden label of a sentence is neutral, the model's output for entailment and contradiction labels should be random across various templates. Conversely, the model should ideally adhere to a consistent label for all templates. Leveraging this assumption, we can map the binary label predictions into the original three labels' by analyzing the model's predictions across different templates. But in practical scenarios, the model is less likely to consistently predict the correct label when the label is entailment and contradiction due to noise and bias. Therefore, judging the model's confidence in its choices or determining whether its predictions are random is challenging based solely on the predictions. In this case, we leverage decision trees as a tool for facilitating the mapping

[2] In the general case, z_i can be a set of words (Z_i) that represents the label l_i.

process. Our approach utilize the model's predictions and their prior accuracy on each templates as features for the decision tree. We can get the final prediction label l using l'_{T_i} and A_{T_i} as follows:

$$l = DT(l'_{T_i}, A_{T_i}), \quad \forall T_i \in \mathbb{T} \tag{2}$$

Here, DT denotes the decision tree, A_{T_i} is the accuracy of each templates and l is the final prediction. This approach allows us to map the pseudo label back to a three-class classification.

4 Experimental Set-up

4.1 Datasets & Templates

We evaluate our method on three Natural Language Inference datasets described below.

SICK. The **S**entences **I**nvolving **C**ompositional **K**nowledge (SICK; Marelli et al., 2014) dataset consists of around 10,000 English sentence pairs, each annotated the entailment relation, including entailment, neutral and contradiction. The distribution of the three relationships is markedly uneven, with 29% sentence pairs annotated entailment, 57% as neutral and only 14% as contradiction.

SNLI. **S**tanford **N**atural **L**anguage **I**nference (SNLI; Bowman et al., 2015) dataset is a widely used benchmark, created by the Stanford NLP group. The dataset is designed to evaluate models' ability to understand and reason about the relationships between pairs of sentences. In contrast to SICK's, biased distribution, each of the three labels in this dataset accounts for approximately 33.3% of the total instances.

MNLI. The **M**ulti-Genre **N**atural **L**anguage **I**nference (**MNLI**) dataset is one part of the GLUE benchmark (Wang et al., 2018). Given a premise sentence and a hypothesis sentence, the task is to predict whether the premise entails the hypothesis contradicts the hypothesis, or neither (neutral). The premise sentences are gathered from ten different sources. An illustration of the distribution of labels for these three datasets is provided in Appendix.

Using these datasets we created 10 original templates such as the one presented earlier in 3.1 or slightly different or templates inspired by other works (Schick and Schütze, 2020). See Appendix for the sum of all templates. We also use expansion tricks to increase the number of templates. See also Appendix for the expanded templates.

4.2 Pre-trained LMs and Evaluation

As previously underlined, the models used were BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b), GPT2 (Radford et al., 2019) and BART (Lewis et al., 2020). All of them are under zero-shot settings and retrieved directly from HuggingFace. Since no training is taking place, we did not needed to adjust hyperparameters such as learning rate or epochs, but for faster processing during inference time we did use batch size of 8. Lastly, regarding the evaluation of our experiments we choose the metric of *Accuracy* over 3 classes.

5 Results

5.1 Main Results

The performance of our methods on the 3 benchmarks are presented in Table 1 (best scores are shown in bold). For SICK and SNLI dataset, we report the accuracy on the test set. For MNLI we report the accuracy on the validation set since their test set isn't publicly available.

Methods	**SICK**	**SNLI**	**MNLI**
baseline$_{majority}$	**56.8**	33.3	33.3
BERT$_{base}$	49.7	34.7	37.3
BERT$_{large}$	41.6	33.0	36.2
RoBERTa$_{base}$	42.0	37.0	36.8
RoBERTa$_{large}$	35.2	**45.0**	**42.9**
GPT2$_{small}$	53.1	38.3	35.2
GPT2$_{large}$	**56.8**	36.0	36.3
BART$_{base}$	**56.8**	40.6	38.7
BART$_{large}$	30.91	43.2	39.3

Table 1: Accuracy (%) of different models on three datasets

Results in Table 1 shows a notable improvement in performance compared to the baseline on SNLI and MNLI dataset when using . These results suggest that the models possess certain knowledge to effectively solve NLI tasks. Furthermore, the LM is more likely to understand the nature of the problem with the help of prompts, without requiring fine-tuning, leading to more accurate predictions. However, our methods lack behind the baseline on SICK dataset. This discrepancy can be attributed to the substantial bias present in SICK, where 56.8% of the dataset labels are tagged as neutral and thus the "baseline" stands at this rate. Meanwhile, the zero-shot method encounters challenges in handling the neutral label.

5.2 Results without Mapping

As mentioned in 3.1, models struggle with explicitly predicting the ”neutral” label. To illustrate the gap in the model's ability across entailment, contradiction, and neutral predictions, we present the accuracy of the RoBERTa$_{large}$ model for these three labels across three datasets in Fig. 1. We set z_3=“unknown” to represent the ”neutral” label.

Despite variance in the accuracy of entailment and contradiction labels of different templates, which may because of the model’s sensitivity to specific templates, the average accuracy remains relatively high. Conversely, the accuracy of the neutral label hovers around zero, verifying the lack of ability for zero-shot models to explicitly predict the “neutral” label with prompts.

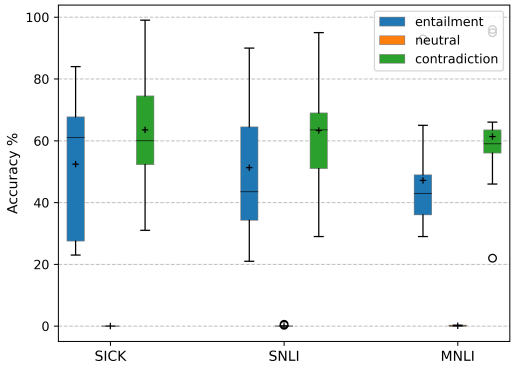

Figure 1: The accuracy of the three labels across the three datasets is assessed using RoBERTa$_{large}$.

6 Analysis

Besides the proposed method earlier we took some more explorative steps which they will give a more concrete hypostasis to our investigation upon solving NLI with prompted MLM techniques.

6.1 Exploiting important tokens

As discussed in sub-section 3.1 a key factor that enables models to discern between the given classes, is to map the labels (l_i) to some word z_i or some set of words Z_i. Here, we try to construct such sets Z_i that they will drive the models towards more plausible results. The main idea of our framework is that given a model M with a vocabulary V and a template T, we avail a small segment of the validation data[3] (S_{val}) and the respective part of the golden labels (G_l) to extract which tokens seem to be important for each label.

More precisely, the output of the model per example i, is the probability distribution over V which we denote as PD_{V_i}. We get the top m probabilities from PD_{V_i} which correspond to the top m most probable tokens-substitutions for the “[MASK]”. Having the aligned golden labels in match, we untangle which $m-$ dimensional set of tokens (x_m) corresponds to which of the 3 labels and we concatenate all sets x_m which account for the same label. As a result, we obtain 3 new sets, V_e, V_n and V_c of lengths $k_e \times m$, $k_n \times m$ and $k_c \times m$. The constants k_e, k_n, k_c represent the absolute frequency of the labels “entailment”, “neutral” and “contradiction” in the set of golden labels G_l. At this point, we can get the pairwise differences between V_e, V_n and V_c as follows:

$$u_e = V_n - V_c, u_n = V_c - V_e, u_c = V_e - V_n \quad (3)$$

Now u_e, u_n and u_c are the sets which contain the tokens that are uniquely present only in one of the labels inside the segment S_{val} and can be used as guided set of words Z_i.

At inference time, instead of using the naive way of getting the token with the highest probability from u_e, u_n and u_c, we use these sets as representatives of the 3 classes respectively and we get the m' most probable tokens per example of the test set[4]. Since this $m'-$ dimensional set of tokens ($x_{m'}$) is sorted we can use it’s indices to retrieve information about which tokens from the 3 abovementioned sets contribute to higher weight of importance per class. Obviously, the lower the index the more significant the token is. If a token from u_e (or u_n or u_c) is absent in $x_{m'}$ then a standard weight is added as penalty.

We can formulate this importance per class as w_e, w_n and w_c respectively and stack them into a 3-element set $w = [w_e, w_n, w_c]$, where each of the w_e, w_n and w_c is given by the following equation:

$$w_{label} = \frac{1}{|u_{label}|}\begin{cases}\sum\limits_{t\in u_{label}} id_t, & \text{if } t \in x_{m'}\\ \sum\limits_{t\in u_{label}} |x_{m'}|, & \text{if } t \notin x_{m'}\end{cases} \quad (4)$$

where t is a token from u_e (or u_n or u_c), id_t denotes the index of the token in $x_{m'}$, $|x_{m'}| = m'$ is the length of the set $x_{m'}$ which we consider to be the penalty in case of absence of the token t from $x_{m'}$ and the subscript “label” can be either e, or n or c for entailment, neutral and contradiction respectively. In the final classification we choose the label whose the weight w_{label} is minimal.

6.1.1 Results from important tokens

We evaluated the method described in sub-section 6.1 only in SNLI and MNLI datasets. We decided

[3] Typically around 1% of the total length.

[4] Note that we can opt if $m = m'$ or $m \neq m'$.

to absent from SICK due to it's uneven distribution of labels. We chose as models BERT$_{base}$ and RoBERTa$_{base}$ and we set the dimensionality of the important tokens set, $m = 100$ for the validation data and $m' = 150$ for the test data. In Appendix we provide a sample of the unique tokens per each of the three labels, found for BERT, under the template t_1 for SNLI. In order to compare the described method we set 2 baselines. The first one, is the majority method which stands approximately at 33.3% for both SNLI & MNLI and the second is the same models when we don't use guided important sets, but we utilize some standard prompts such as: Z_1={ 'true', 'yes', 'right'}, Z_2={'false', 'no', 'wrong'} and Z_3={ 'unknown', 'other', '.'} mapped to entailment, contradiction and neutral correspondingly [5].

Fig. 2 illustrates the results from this experimentation using *accuracy*, as a metric on SNLI (test set) and MNLI (validation set). In this figure, "model_classical" refers to when we don't use guided sets u_{label} and "model_guided" when we do use. Regarding SNLI we can see that for both BERT and RoBERTa the usage of guided sets of tokens (orange solid line) aids them to increase their score and in the most cases overpass the majority classifier (gray dashed line), in 8 out of 10 templates for both models. On the contrary, on MNLI dataset it's not so clear if this method is preferred for BERT over the majority pick, but undoubtedly boosts it's performance from when using unguided tokens. Nonetheless, for RoBERTa we observe a robust outcome above 33.3% on 9 out of 10 times and simultaneously at least a 5% improvement of the accuracy over the unguided model in average.

6.2 The effect of the sequence length

One question that arises when dealing with LMs and their ability to understand natural language, is whether the length of the input sequence affects the output of the model and consequently it's accuracy on predictions. Considering a sentence j, a *premise* of length L_{p_j}, such as $p = (t_1, t_2, \ldots, t_{L_{p_j}})$ and a *hypothesis*, $h = (t_1, t_2, \ldots, t_{L_{h_j}})$ of length L_{h_j}, where t_k are the tokens present on the premise and the hypothesis, then we can define the total length of the sentence as ($L_j^{T_i}$) within the template i with the following equation:

$$L_j^{T_i} = L_{p_j} + L_{h_j} + L_{b_{T_i}} \quad \forall T_i \in \mathbb{T} \tag{5}$$

where $L_{b_{T_i}}$ is the length of the basis of the template used. All three of L_{p_j}, L_{h_j} and $L_{b_{T_i}}$ can vary depending on the lengths of premises, hypotheses and the basis respectively. For example, the basis of the original template 9 is shorter with $L_{b_{T_9}} = 4$ tokens, than the basis of original template 2 with $L_{b_{T_2}} = 9$ tokens (see Appendix).

Taking this into account, we found the average length of instances for all of the 10 original templates used (denoted as $\mu_{L_{T_i}}$) and then we segregated the instances with length less than $\mu_{L_{T_i}}$ as "short-examples" (Sh_i) and those with higher length than $\mu_{L_{T_i}}$ as "long-examples" (Lo_i). More formally, we say that for each template i we have:

$$j \in \begin{cases} Sh_i & \text{if } L_j^{T_i} \leq \mu_{L_{T_i}} \\ Lo_i & \text{if } L_j^{T_i} > \mu_{L_{T_i}} \end{cases} \tag{6}$$

where $i \in \{1, \ldots, 10\}$. Our purpose is to examine whether the models' accuracy differs between testing on set Lo_i and Sh_i. We except that in most cases the accuracy will be diminished or uninfluenced, rather than increased.

6.2.1 Results with different sequence length

Like in 6.1.1 the evaluation of the previous method was carried out, only in SNLI and MNLI. Once more, we didn't use a classification head in the top of the models (this time we applied all those mentioned in sub-section 4.2), but we utilized the same standard prompt sets Z_1, Z_2 and Z_3 from 6.1.1. The results from this experiment regarding the *accuracy* of the models over the 10 original templates are shown in Fig. 3, for SNLI and MNLI. The reported accuracy scores are rounded to the nearest 0.5% for clarity.

As seen from Fig. 3 for SNLI, we can vividly postulate that the length of examples affects the models' accuracy negatively especially for BERT and BART, as we can observe that the blue solid line is always greater or equal to the light-blue and dashed one. For RoBERTa and GPT2 the situation is more vague, but still a slight deterioration can be distinguished when moving from shorter to longer examples. Coming to MNLI results of Fig. 3, we can perceive similar behaviour. However, this time its clear that GPT2 and BART models exhibit a discernible decline in accuracy as sequences transition from shorter to longer lengths. Conversely, the behavior of BERT and RoBERTa appears more nuanced as we can't say clearly whether lengthy sequences improved or impaired their accuracy.

[5] Note that here we didn't use classification head (prediction mapper) like in our main method in3.2

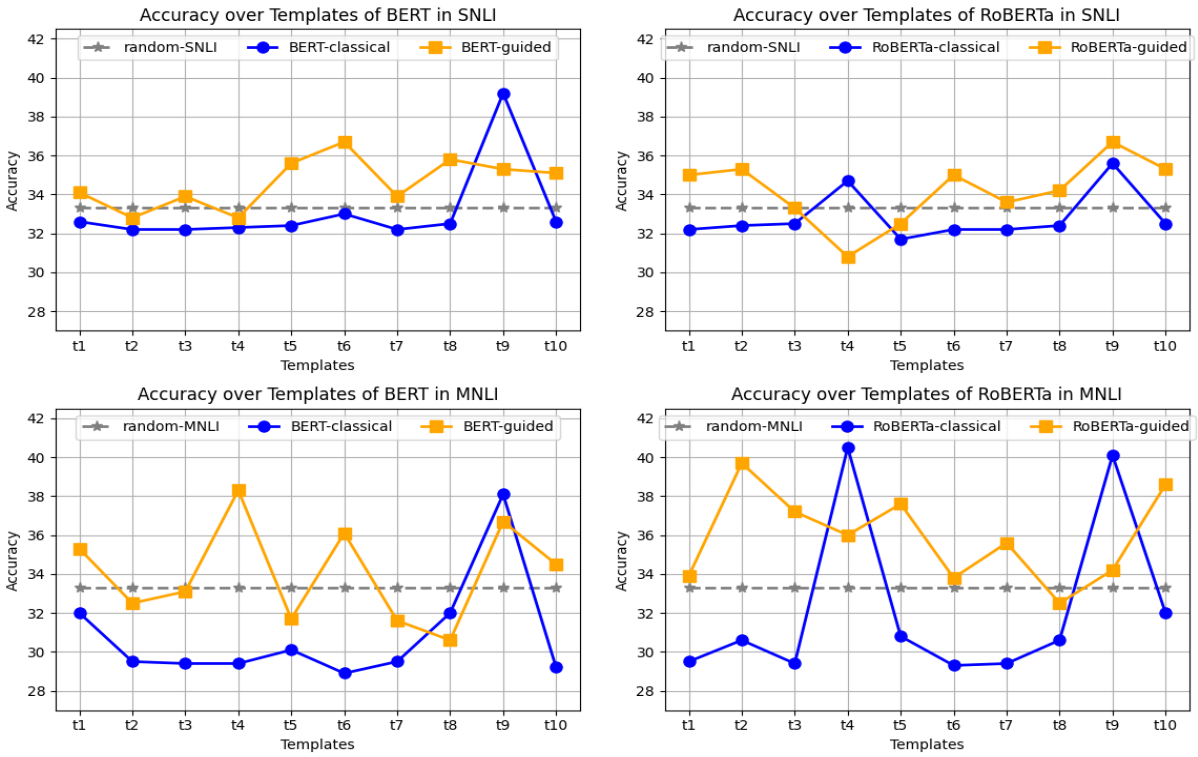

Figure 2: BERT & RoBERTa accuracy over the 10 templates, tested with and without important token-sets in SNLI & MNLI

6.3 Differences Between Prompts

As observed in Fig. 1, a disparity exists between the performance of the best and worst prompts. To gain insights into the insights of these prompts, we performed an analysis using SNLI and RoBERTa$_{large}$ as an example. The best-performing prompt is:

When p is true, then h is [MASK].

Notice that the result is based on the whole dataset, so the result might be a bit surprise when having seen Figure 3. This prompt achieves a high overall accuracy of 67% accuracy (72% for entailment and 62% for contradiction label). This outcome is reasonable, as the prompt adopts a common linguistic structure frequently appeared in human speakings when judging the relationship between sentence pairs. Consequently, it is likely that the LM has encountered similar sentence forms during its training on the corpus. Interestingly, the worst-performing prompt is an expansion of the best one:

When h is false, then p is [MASK].

This finding is foreseeable and can be explained. Previous works (Kassner and Schütze, 2020) have proved the limited ability of language models to handle negative words (e.g., ”not”, ”no”) in a sentence. Moreover, there are instances where the order between premise and hypothesis matters. Given these factors, a decrease in performance is acceptable for this prompt. Additionally, some prompts have almost 100% accuracy in one label but very low accuracy in the other. For instance:

When the premise ’p’ is: true,
then the hypothesis ’h’ is: [MASK]

It is evident that models didn’t catch this prompt. Therefore, we simply exclude the results based on these prompts during the mapping procedure.

7 Conclusion

We proposed a zero-shot prompt-based method for NLI tasks. Our approach initially uses prompts to transform the NLI task into a MLM task, predicting the entailment/contradiction relationship between sentence pairs. Subsequently, a decision tree is employed to obtain the final prediction. Our method demonstrates notable performance on two prominent NLI benchmarks. We also present experimental results on various pre-trained LMs and explore the performance differences of models under different templates. Additionally, we delve into the model’s prompt-handling capabilities, discussing the properties of important tokens and investigating whether the sequence length affects performance. Our future work will focus on improving prompts so as to models accomplish better results and evaluate our methods on more NLI datasets.

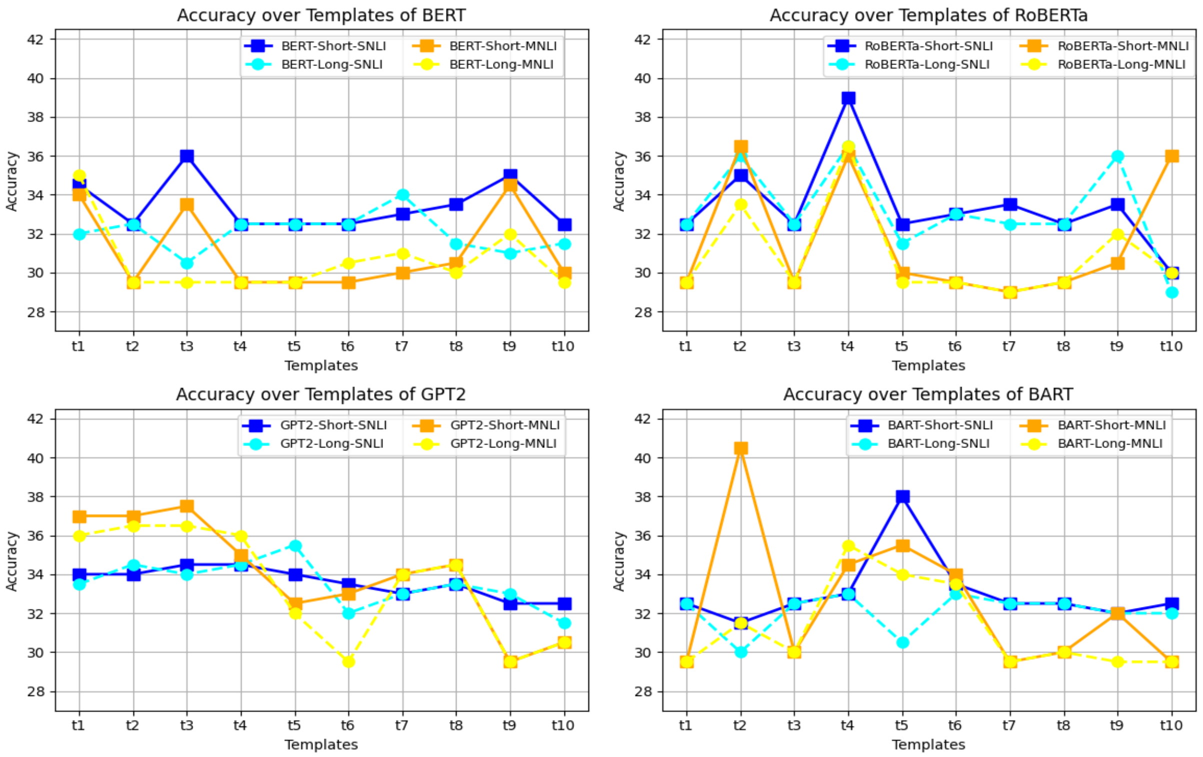

Figure 3: Models accuracy over the 10 templates when tested on the derived Sh_i and Lo_i sets of SNLI & MNLI

8 Contributions

The contributions for this paper are as follows:

- **Wenkai Chen**: Abstract, Related Work, Methods, Experimental Set-up, Results, Analysis 6.3, Conclusion, Appendix $A.2$, $A.3$, and code that correspond to the listed sections.
- **Admitos Passadakis:** Introduction, Experimental Set-up, Analysis 6.1, 6.2, Appendix $A.1$, $A.4$, References, Revisions of the paper and code that correspond to these sections.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020a. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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. 2020b. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

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.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert:

Improving pre-training by representing and predicting spans. *CoRR*, abs/1907.10529.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7811–7818, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2019. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, abs/1910.13461.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. 2022. Language models of code are few-shot commonsense learners.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few-shot text classification and natural language inference. *CoRR*, abs/2001.07676.

Timo Schick and Hinrich Schütze. 2021. 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.

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, Brussels, Belgium. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *CoRR*, abs/1704.05426.

A Appendix

A.1 Labels Distribution

As underscored in sub-section 4, here we provide a depiction of the distribution of labels among SICK, SNLI and MNLI datasets.

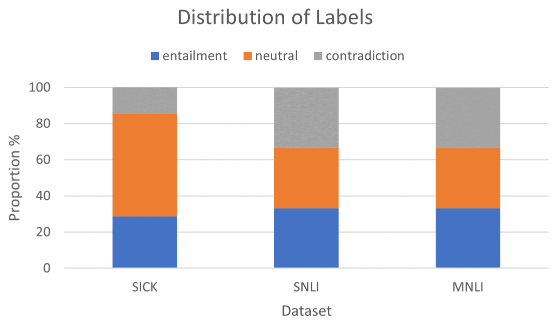

Figure 4: Distribution of labels in SICK, SNLI and MNLI datasets.

A.2 Prompts

Table 2 shows the original prompts (templates) we used in our methods, where p stands for *premise* and h for *hypothesis*.

A.3 Expanded Prompts

Inspired by the data augmentation, we reverse the order of premise and hypothesis to expand the prompts. Table 3 shows one such example for prompt expansion derived from template 1 of Table 2. The expanded prompts are used as features in the decision tree.

	Prompt	Entailment	Contradiction
1	if p is 'true', the h is '[MASK]'	true	false
2	Suppose that p is: true, then h is: [MASK]	true	false
3	When p is true, then h is [MASK]	true	false
4	When the premise 'p' is: true, then the hypothesis 'h' is: [MASK]	true	false
5	When the premise p is true, then the hypothesis h is: [MASK]	true	false
6	Considering that the premise 'p' is 'true', then the hypothesis 'h' is: '[MASK]'	true	false
7	Knowing that the premise 'p' has a label 'true', then the hypothesis 'h' will have a label '[MASK]'	true	false
8	Regarding that the premise 'p' is: 'true', then the hypothesis 'h' will be '[MASK]'	true	false
9	p: true, h: [MASK]	true	false
10	p? [MASK], therefore h	Yes	No

Table 2: Original prompts that are used in our method.

label	Prompt	Entailment	Contradiction
original	if p is 'true', the h is '[MASK]'	true	false
expansion	if h is 'true', the p is '[MASK]'	true	false
expansion	if p is 'false', the h is '[MASK]'	false	true
expansion	if h is 'false', the p is '[MASK]'	false	true

Table 3: An example of prompt expansion.

A.4 Top unique tokens per label

As mentioned earlier in 6.1 during our search for the tokens that uniquely appear in each label we deduced some sets that contain those tokens. These guided sets were u_e, u_n and u_c for the entailment, neutral and contradiction labels respectively. Here we depict the content of those sets with a parameter $m = 100$ used for the validation set of the SNLI dataset, for the BERT model, under the template t_1. We should underscore that lengths of these sets originally were 22 for entailment, 14 for neutral and 28 for contradiction, meaning that for that specific segment of S_{val} the tokens that were uniquely presented in each of the three labels were 22, 14 and 28 correspondingly. Nevertheless, for visualization reasons, here we illustrate only the top-10 from each of the three label categories, in Fig. 5.

sorry	offensive	fun
imagination	dangerous	pretty
laughter	loose	sense
stories	proved	unexpected
implies	consistent	.
blue	non	this
terrible	crazy	solid
humorous	w	are
amusing	high	music
broken	win	remarkable

(a) Top-10 tokens only in Entailment | (b) Top-10 tokens only in Neutral | (c) Top-10 tokens in Contradiction

Figure 5: Top-10 unique tokens for each of three different label categories.