Probing Language Models for Commonsense Knowledge using Template Variations

Jesse Dodge[♠] **Karishma Mandyam**[♡] **Akari Asai**[♡]
Hannaneh Hajishirzi[♡♠] **Noah A. Smith**[♡♠]
[♡]Paul G. Allen School of Computer Science & Engineering, University of Washington
[♠]Allen Institute for AI
jessed@allenai.org

Abstract

Probing language models directly after pretraining (without fine-tuning) has helped separate the kinds of information learned in the two training stages. In this work, we build a probe dataset of "fill-in-the-blank" cloze-style statements designed to test a model's commonsense reasoning ability. While probes of factual knowledge can be answered by retrieving correct answers from existing text, our probe requires fundamentally different types of reasoning, such as reasoning about the emotional state of a person taking an action. We find that performance on this task varies dramatically across ten pretrained language models, with BERT and RoBERTa outperforming more recent models such as T5. We also find that model performance varies significantly with small changes to the probes (such as including a period at the end or not), suggesting that language models are brittle to semantics-preserving perturbations.

1 Introduction

Language models have become ubiquitous in natural language processing (Devlin et al., 2019; Liu et al., 2019). The diversity of tasks which have seen improvements from this paradigm suggests that pretraining may imbue language models with some general language processing abilities; to understand the impact of pretraining we can analyze the model before fine-tuning. One format for such probes is cloze-style (Petroni et al., 2019; Jiang et al., 2020) fill-in-the-blank statements. Here, a model must correctly predict the masked token (typically the highest scoring in its vocabulary). Probes in this format are easily understandable and align with masked token prediction training.

Probing for factual knowledge can highlight a model's retrieval capabilities, but we may want to analyze different phenomena such as

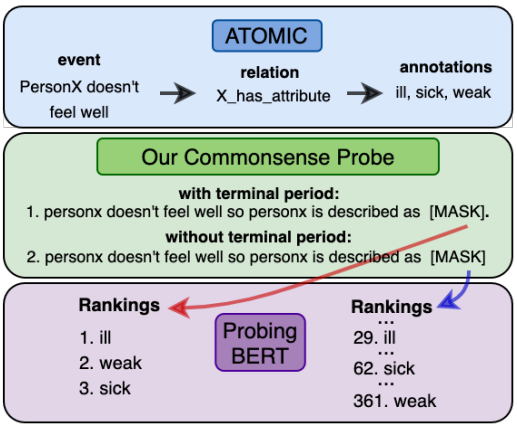

Figure 1: An example of how an event-relation-attribute triple in ATOMIC is used to generate our commonsense probe. We show two of our eight variants, here varying only the terminal punctuation. The ranks of the correct predictions drop dramatically without a final period, revealing brittleness to even minor changes.

commonsense reasoning, which differs from factual knowledge in a number of ways. For instance, given a textual description of some interaction between two people, humans can reason about plausible motivations for or effects of their interaction. While factual knowledge can often be found written in informational texts, commonsense knowledge typically isn't written down, making this a challenging task.[1] Moreover, with commonsense probes, multiple answers can be equally plausible, with only a few annotated in existing knowledge bases (KBs).

In this paper we introduce a cloze-style commonsense probe dataset built from ATOMIC (Sap et al., 2019), a commonsense KB, in which each example takes the form of a sentence with a

[1]This is often referred to as "reporting bias" (Gordon and Durme, 2013). The bigram "black sheep" appears much more frequently than "white sheep", even though it is common knowledge that sheep are typically white.

single masked token (See Figure 1). We construct the probe dataset by carefully designing templates which convert the commonsense relations into natural language, and explore variations of these templates by introducing semantics-preserving perturbations such as casing or adding a terminal period.

We evaluate ten pretrained language models on this task, predicting the masked token in each sentence, and find a wide range of performance. We show that language model performance varies significantly with minor perturbations to the structure of the probing sentences (such as including a period at the end vs. not). This suggests that language models are brittle to prompt perturbations that should not matter, like capitalization and certain punctuation, and that to effectively query these models for commonsense knowledge, we must carefully design the probes.

2 Constructing Probes from ATOMIC

ATOMIC captures commonsense relations for everyday events, causes, and effects. Each event revolves around one or more characters (PersonX, PersonY, and so forth). ATOMIC consists of nine *if-then* relations, and the data can be "viewed" in two different ways (Persona-Event-MentalState or Causes-Stative-Effects) as prescribed by Sap et al. (2019). Each example *i* in ATOMIC consists of an event e_i and one or more annotated attributes $a_i^{r_j}$, expressed as free text, for each *if-then* relation r_j. As an example, if e_i="PersonX repels PersonY's attack" and r_j="PersonX feels", then $a_i^{r_j}$ could be "angry" or "tired". Table 4 in the appendix specifies the two views. We construct our challenge dataset from the ATOMIC development data, and Table 1 specifies the statistics for each relation.

Constructing templates For each (e_i,r_j) pair we use a template to generate a sentence using e_i as a prefix and r_j as a suffix containing the mask token, where the correct answer for the masked token is the associated gold attribute $a_i^{r_j}$. Continuing the example above, the probe is "PersonX repels PersonY's attack. As a result, PersonX feels [MASK]." Correct answers for the [MASK] token include "angry" and "tired". See Table 3 for the relation-specific suffixes. Following Petroni et al. (2019), we constrain all target attributes to be one word.

Filtering ATOMIC data We *exclude* the following types of data in ATOMIC when creating our dataset: a) the annotated attribute is missing, b) the annotated attribute is explicitly "none", c) the event e_i contains blank spaces or underscores.[2]

Type	ATO. Ex.	Filt. Ex.	Eval Ex.	Uniq. Ex	Avg Lab.
Stative	12952	8901	6321	1677	3.8
Causes	16571	337	299	251	1.2
Effects	50077	7513	6493	3233	2.0
Event	47218	1496	1181	992	1.2
Mental State	19430	6354	5611	2492	2.3
Persona	12952	8901	6321	1677	3.8
Total	159200	33502	13113	5161	2.5

Table 1: Dataset statistics for both views in our commonsense benchmark. **ATO. Ex.**, **Filt. Ex.**, **Eval Ex.**, **Uniq. Ex** denote the number of the original event-attribute examples in ATOMIC, the number of examples after filtering, the number of examples evaluated (using our vocabulary) and the number of unique examples after aggregating annotations, respectively.

Multiple annotations Unlike factoid question answering or relational databases, the nature of commonsense knowledge bases is that there can be many correct answers. If there are multiple annotated attributes for an event and a relation, we retain all of the attributes, denoting each one of them as equally possible gold answers.

Grammaticality Grammatical errors affect models' behavior, and robustness against those errors varies across existing language models (Yin et al., 2020). To improve the grammaticality of our dataset, we manually annotate each attribute $a_i^{r_j}$ with a POS tag, which we combine with the relation to generate a grammatical template. Given the relation 'xWant' and a noun attribute, our template generates 'As a result, PersonX wants [MASK]', but with a verb attribute, we generate 'As a result, PersonX wants *to* [MASK]'. More details of this process are in the appendix.

Variations on Cloze-Style Templates We construct simple variants of the templates used to generate the challenge data. While previous work has shown that mining for template variants with different syntax and semantics can lead to improved performance on cloze-style tasks (Jiang et al., 2020), we examine the impact of only changing punctuation and case, which should be essentially semantics-preserving. Specifically, we evaluate three ways of varying our templates: **case** (cased vs. uncased), **the number of sentences** (two sentences vs. one sentence with two clauses con-

[2] In ATOMIC blank spaces represent arbitrary arguments for a verb in an event. Since these are unspecified, we ignore these examples in an effort to construct grammatical prompts that will be familiar to a language model.

joined with "and" or "so"), and **a terminal** period (a period after [MASK] vs. no period after [MASK]). Table 5 in the Appendix describes the transformations for each template.

3 Models and Evaluation

We evaluate ten pretrained language models on all eight sets of templates. These include Transformer-XL (Dai et al., 2019), ELMo-base and ELMo-5B (Peters et al., 2018), BERT-base and BERT-large (Devlin et al., 2019), DistilBERT (Sanh et al., 2019), RoBERTa-base and RoBERTa-large (Liu et al., 2019), and T5-base and T5-large (Raffel et al., 2019). We use cased versions of all models. The models are *not* fine-tuned or modified in any way prior to evaluation. We use a common vocabulary of 18k case-sensitive tokens (the intersection of the vocabularies of all ten models, following Petroni et al. (2019)). All experiment reproducibility information can be found in the Appendix.

Evaluation methods The annotations in ATOMIC for a given instance represent a *subset* of the annotations which would be considered correct by a reader. This is the nature of the commonsense task. It could be the case that our model predicts a reasonable, but not annotated, response. Thus, our results are a lower bound on true performance and we evaluate P@K curves. To account for multiple annotations in ATOMIC (see Table 1), we calculate Precision@K (**P@K**) for an example as 1 if *any* of the gold annotations are in the top K predictions.[3]

4 Experimental Results and Analysis

We see overall performance in Figure 2. With 18k tokens in our vocabulary random guessing P@10 would be 10/18k. Surprisingly, our best models (BERT-large and RoBERTa-large), not fine-tuned on this task, observe P@10 of almost 30%.[4] However, DistilBERT, T5-base, and T5-large all have very low scores, with even P@100 at less than 3%. As noted by Jiang et al. (2020), with these probing tasks, results on a set of templates can only provide a *lower bound* on the actual knowledge in the models and a different prompt may result in more accurate predictions.

[3] Although with this setup, more annotations per example can lead to higher scores, all else equal, we empirically found that the difficulty of the tasks is the primary influence on precision, not the number of annotations.

[4] seq2seq models trained on the full ATOMIC data yield around 45% P@10 (Sap et al., 2019), though our data consists of only single-token attributes from ATOMIC so is not comparable.

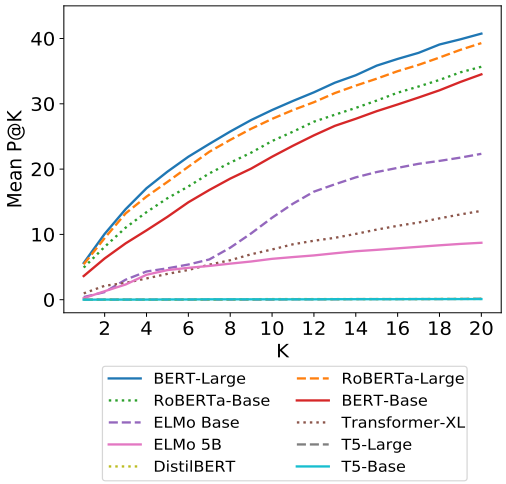

Figure 2: P@K (%) with K between 1 and 20, for all ten models evaluated, on the best template set (uncased, one sentence, with period at the end).

Figure 3 shows per-relation P@K results for the best performing two models, BERT-large and RoBERTa-large. Both models understand the Mental-State and Causes relations of ATOMIC best. These two relations primarily capture the intents and reactions of PersonX, suggesting that models can reason better about the implicit motivations of the subjects in a sentence. The Stative and Persona relations (which capture the same ATOMIC relation "X attribute") perform relatively worse. These relations have the most average annotations per example, suggesting that they are more open ended. Model predictions for these relations might be valid but simply not annotated in our dataset.

Table 2 presents P@10 for the best two models, BERT-large and RoBERTa-large, as we vary the templates. We include the full P@10 and P@20 tables for all models in Table 7 and Table 8 in the Appendix. Best-found performance for BERT (29.3%) is slightly higher than RoBERTa (28.6%). However, RoBERTa is more stable to the variation in these templates; the worst performance for RoBERTa is 19.3% while BERT is 8.0%.

Case BERT and RoBERTa respond differently to changing the case; BERT performs better with the cased templates, while RoBERTa has slightly stronger performance with 3/4 of the uncased templates. It's possible when we process sentences with proper nouns, casing might help the models, but with our commonsense prompts which don't have many proper nouns uncased is better.

Period Adding a terminal period positively affects both BERT and RoBERTa models the most. On

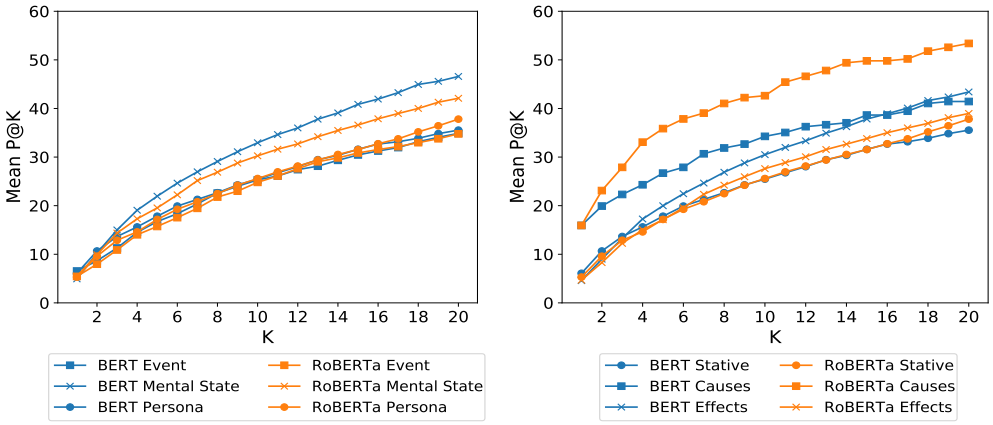

Figure 3: P@K curves for BERT-large and RoBERTa-large (the best two performing models), on the best (P@20) template set (uncased, one sentence, with period at the end), split by the Event-Mental State-Persona (left) and Stative-Causes-Effects views of the ATOMIC data.

Template	BERT	RoBERTa
Case-Period-One	29.3	28.6
Case-Period-Two	26.6	25.2
Case-No Period-One	11.5	20.7
Case-No Period-Two	8.4	19.3
Uncase-Period-One	29.0	27.7
Uncase-Period-Two	23.9	25.9
Uncase-No Period-One	8.2	20.8
Uncase-No Period-Two	8.0	19.8

Table 2: P@10 (%) for BERT-large and RoBERTa-large, the two best performing models as we vary the templates. P@10 and P@20 results for all ten models are the appendix.

average, BERT-large gains 18.2 P@10 points while RoBERTa-large gains 6.7 P@10 points. The presence of a period indicates sentence boundaries, and therefore it might help to know about which words generally come at the end of the sentences.

Number of sentences For nine out of ten models, when we interpret the prompt as two sentences instead of one, the performance decreases by at most 3 P@10 points on average. While performance doesn't decrease dramatically, one explanation for this phenomenon might be that models have trouble capturing the context of the first sentence while predicting the [MASK] token. Understanding the event is critical to predicting the correct token, and separating these two parts into different sentences could confuse the model.

5 Related Work

Prior work on formulating probing tasks from existing knowledgebases has demonstrated that language models store some factual knowledge (Petroni et al., 2019; Xiong et al., 2020). By fine-tuning models to answer factoid questions, large pretrained models have achieved competitive performance without any access to existing context (Roberts et al., 2020; Lewis et al., 2020). In contrast to works like Bosselut et al. (2019), which train GPT-2 on ATOMIC as knowledge base completion, we focus on probing for *inferential commonsense* without further training, evaluating knowledge which is rarely stated explicitly in existing corpora (Zhang et al., 2017).

Jiang et al. (2020) build on the work of Petroni et al. (2019) to investigate the effects of rephrasing probes on knowledge probing tasks. They propose methods to construct prompts which achieve higher accuracy on relational knowledge benchmarks, demonstrating that template design plays an important role in retrieving factual knowledge. This technique changes the prompt in more significant ways, while our work suggests that smale semantics-preserving changes could also significantly affect the models' performance.

6 Conclusion

In this work, we introduce a new commonsense probing task based on Sap et al. (2019) to test pretrained language models' knowledge of inferential commonsense knowledge. We explore constructing semantic-preserving variants of our probes to see how small perturbations affect model performance in a zero-shot setting. Among 10 language models, BERT and RoBERTa significantly ouperform others. We show that slightly perturbing the probe drastically affects model performance, suggesting that our results are a lower bound and that models

are brittle to minor changes in the prompt.

References

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

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.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge acquisition. In *Workshop on Automated Knowledge Base Construction (AKBC)*.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics (TACL)*, abs/1911.12543.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.

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. *arXiv preprint arXiv:1907.11692*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *EMNLP-IJCNLP*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3027–3035.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. *ArXiv*, abs/1912.09637.

Fan Yin, Quanyu Long, Tao Meng, and Kai-Wei Chang. 2020. On the robustness of language encoders against grammatical errors. In *arXiv preprint arXiv:2005.05683*.

Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2017. Ordinal common-sense inference. *Transactions of the Association for Computational Linguistics*, 5:379–395.

A Details on dataset creation

In this section, we provide the examples and more details on our dataset collections.

A.1 ATOMIC Relation Groupings

We show the relation groups defined by Sap et al. (2019) in Table 4.

A.2 Templates for each relationship type in ATOMIC

Table 3 show the list of templates for all relation type in ATOMIC. As discussed in Section 2, we append those templates to create our probing task.

Relation	Suffix
X intent	This is because PersonX wanted [MASK]
X need	In order to do this, PersonX needs [MASK]
X attribute	PersonX is described as [MASK]
Effect on X	As a result, PersonX [MASK]
X want	As a result, PersonX wants [MASK]
X reaction	As a result, X feels [MASK]
Other reaction	As a result, others feel [MASK]
Other want	As a result, others want [MASK]
Effect on other	As a result, everyone else [MASK]

Table 3: Suffixes appended to event text data for each *if-then* relation in ATOMIC.

A.3 Grammaticality

The process of constructing the ATOMIC dataset led to some noise, so to improve the grammaticality of our dataset we manually cleaned the data by searching through the most frequent labels and removing extraneous punctuation ('happy' vs. 'happy.'), capitalization ('smiles' vs. 'Smiles'), and typos ('noticable' becomes 'noticeable'). We then manually annotated a list of approximately 700 labels[5] with a coarse part of speech tag: noun, verb, or adjective. We use the POS tag in combination with the relation to generate a template. Given a relation 'xWant' and a noun attribute, our template generates 'As a result, PersonX wants [MASK]', but when given a verb attribute, 'As a result, PersonX wants *to* [MASK]'.

A.4 Varying Prompt Templates

We illustrate how we add punctual perturbations create semantic-preserving examples with an event "PersonX is going on a camping trip" in Table 5. Those changes do not change the semantic of the original examples.

[5]These were all attributes mentioned in six of the nine relations in ATOMIC (oEffect, oWant, xEffect, xIntent, xNeed, and xWant). The remaining three relations were primarily comprised of labels with a single POS tag.

B Reproducibility Materials

We will release the code for our paper upon publication. Table 6 highlights the number of parameters in each evaluated model. Table 6 also describes the average run time per model per template. We ran all models on a GeForce GTX 1080 GPU and a Titan XP GPU. All versions of models trained in this work are from Petroni et al. (2019) or the HuggingFace repository.

C More results

In this section, we provide complete results of our experiments across 10 language models.

C.1 Precision@{10,20} for all the evaluate models on varied templates

We provide the Precision@10 and Precision@20 results for all evaluated models in Table 7 and Table 8, respectively. We also show the average difference in Precision@10 and Precision@20 across models in Table 9 and Table 10, respectively.

Case BERT-large, DistilBERT, and both T5 models perform better when interpreting the input as cased. This trend persists when evaluating P@20 scores. As shown in Table 9 and Table 10, RoBERTa-large is the most robust model while RoBERTa-base is least robust to casing differences.

Period Adding a period at the end of the prompt almost always increases model performance. The only exception is the Transformer-XL model, for which the presence of a period makes no difference.

Number of Sentences For nine out of ten models, when we interpret the prompt as two sentences instead of one, the performance decreases. The only exception is with DistilBERT, where performance increases by 0.1 P@10 points or doesn't change for P@20. For the 9 models that do worse on the two sentence templates, the decrease in performance is usually at the minimum 1 P@10 point or 2 P@20 points.

Category	Included Relations
Stative	X attribute
Causes	X intent, X need
Effects	Effect on X, X want, X reaction, Other reaction, Other want, Effect on other
Event	X need, Effect on X, X want, Other want, Effect on other
Mental State	X intent, X reaction, Other reaction
Persona	X attribute

Table 4: Relation groupings for the Causes-Effects-Stative and Persona-Event-MentalState partitions describing two views of ATOMIC as recommended by (Sap et al., 2019).

Template	Input Sequence
Case-Period-One	PersonX is going on a camping trip and as a result, PersonX feels [MASK].
Case-Period-Two	PersonX is going on a camping trip. As a result, PersonX feels [MASK].
Case-No Period-One	PersonX is going on a camping trip and as a result, PersonX feels [MASK]
Case-No Period-Two	PersonX is going on a camping trip. As a result, PersonX feels [MASK]
Uncase-Period-One	personx is going on a camping trip and as a result, personx feels [MASK].
Uncase-Period-Two	personx is going on a camping trip. as a result, personx feels [MASK].
Uncase-No Period-One	personx is going on a camping trip and as a result, personx feels [MASK]
Uncase-No Period-Two	personx is going on a camping trip. as a result, personx feels [MASK]

Table 5: An illustration of how we vary the input sequence based on the template for a given example. Here, one of the targets for the [MASK] token is 'bored'. Note that to construct the two sentences vs. one sentence template, we employ the connective word 'so'.

Model	Model Parameters	Average Runtime per Template
Transformer-XL	257M	30
ELMo-base	93.6M	25
ELMo5B	93.6M	25
BERT-base	110M	21
BERT-large	340M	21
DistilBERT	66M	21
RoBERTa-base	125M	21
RoBERTa-large	355M	22
T5-base	220M	21
T5-large	11B	21

Table 6: Number of parameters in each evaluated model and Average model runtime (in minutes) for a single template. This sums the runtimes for each of the six relations (stative, causes, effects, event, mental state, and persona).

Template	T-XL	E-b	E-5.5B	BERT-b	BERT-l	D-BERT	RoBa-b	RoBa-l	T5-b	T5-l
Case-Period-One	**7.7**	7.8	5.8	20.1	**29.3**	0.3	20.9	**28.6**	**0.6**	**0.4**
Case-Period-Two	4.6	5.3	4.4	16.8	26.6	**0.4**	19.8	25.2	0.0	0.2
Case-No Period-One	**7.7**	3.1	6.5	10.0	11.5	0.0	15.8	20.7	0.1	0.1
Case-No Period-Two	4.6	1.9	5.4	7.0	8.4	0.0	11.5	19.3	0.0	0.0
Uncase-Period-One	**7.7**	**12.6**	6.3	**21.9**	29.0	0.0	**24.3**	27.7	0.0	0.0
Uncase-Period-Two	5.3	6.1	5.7	19.2	23.9	0.2	22.9	25.9	0.1	0.0
Uncase-No Period-One	**7.7**	4.2	**7.0**	9.4	8.2	0.1	19.6	20.8	0.0	0.3
Uncase-No Period-Two	5.3	2.5	6.5	8.2	8.0	0.1	16.2	19.8	0.1	0.0

Table 7: Precision@10 (%) for all the evaluated models as we vary the templates. Bolded numbers indicate the best performing template for each model.

Template	T-XL	E-b	E-5.5B	BERT-b	BERT-l	D-BERT	RoBa-b	RoBa-l	T5-b	T5-l
Case-Period-One	**13.6**	17.1	8.6	31.4	40.4	**0.6**	31.9	**39.9**	**1.1**	**0.6**
Case-Period-Two	9.1	6.6	6.8	26.5	37.8	**0.6**	30.4	36.6	0.0	0.3
Case-No Period-One	**13.6**	7.8	8.6	18.7	18.7	0.1	24.3	31.1	0.3	**0.6**
Case-No Period-Two	9.1	4.4	6.7	12.9	16.1	0.1	18.9	29.4	0.3	0.0
Uncase-Period-One	**13.6**	**22.3**	**8.7**	**34.5**	**40.7**	0.1	**35.7**	39.3	0.1	0.2
Uncase-Period-Two	10.0	14.5	8.6	30.1	35.1	0.3	34.7	36.9	0.2	0.1
Uncase-No Period-One	**13.6**	9.0	8.5	18.2	16.3	0.3	29.2	31.2	0.0	0.4
Uncase-No Period-Two	10.0	5.9	8.0	14.4	15.0	0.1	25.9	30.6	0.3	0.1

Table 8: Precision@20 (%) for all the evaluated models as we vary the templates. Bold numbers indicate the best performing template for each model.

Template	T-XL	E-b	E-5.5B	BERT-b	BERT-l	D-BERT	RoBa-b	RoBa-l	T5-b	T5-l
Avg (Cased - Uncased)	-0.4	-1.8	-0.8	-1.2	1.7	0.1	-3.8	-0.1	0.1	0.1
Avg (Period - No Period)	0.0	5.0	-0.8	10.8	18.2	0.2	6.2	6.7	0.1	0.1
Avg (One Sent. - Two Sent.)	2.7	3.0	0.9	2.5	2.8	-0.1	2.5	1.9	0.1	0.1

Table 9: Each row indicates average difference in Precision @10 (%) for each variance over all templates. For example, "Avg (Cased - Uncased)" takes the differences between each cased template and its counterpart uncased template and averages them. Negative numbers indicate that the second of the two variations ('Uncased', 'No Period', 'Two Sentence') are better performing and positive numbers indicate the opposite.

Template	T-XL	E-b	E-5.5B	BERT-b	BERT-l	D-BERT	RoBa-b	RoBa-l	T5-b	T5-l
Avg (Cased - Uncased)	-0.5	-4.0	-0.8	-1.9	1.5	0.1	-5.0	-0.2	0.3	0.2
Avg (Period - No Period)	0.0	8.4	0.2	14.6	22.0	0.2	8.6	7.6	0.1	0.0
Avg (One Sent. - Two Sent.)	4.0	6.2	1.1	4.7	3.1	0.0	2.8	2.0	0.2	0.3

Table 10: Each row indicates average difference in Precision @20 (%) for each variance over all templates. For example, "Avg (Cased - Uncased)" takes the differences between each cased template and its counterpart uncased template and averages them. Negative numbers indicate that the second of the two variations ('Uncased', 'No Period', 'Two Sentence') are better performing and positive numbers indicate the opposite.