Towards a Language Model for Temporal Commonsense Reasoning

Mayuko Kimura[1] **Lis Kanashiro Pereira**[2] **Ichiro Kobayashi**[3]
Ochanomizu University, Japan
[1,3]{g1720512,koba}@is.ocha.ac.jp
[2]kanashiro.pereira@ocha.ac.jp

Abstract

Temporal commonsense reasoning is a challenging task as it requires temporal knowledge usually not explicitly stated in text. In this work, we propose an ensemble model for temporal commonsense reasoning. Our model relies on pre-trained contextual representations from transformer-based language models (i.e., BERT), and on a variety of training methods for enhancing model generalization: 1) multi-step fine-tuning using carefully selected auxiliary tasks and datasets, and 2) a specifically designed temporal task-adaptive pre-trainig task aimed to capture temporal commonsense knowledge. Our model greatly outperforms the standard fine-tuning approach and strong baselines on the MC-TACO dataset.

1 Introduction

Although recent pre-trained language models such as BERT (Devlin et al., 2019) have achieved great success in a wide range of natural language processing (NLP) tasks, these models may still perform poorly on temporal reasoning scenarios. Ribeiro et al. (2020) has shown that such models often fail to make even simple temporal distinctions, for example, to distinguish the words *before* and *after*, resulting in degraded performance. An especially challenging task is temporal commonsense reasoning. For instance, given two events "going on a vacation" and "going for a walk", while most humans would know that a vacation is typically longer and occurs less often than a walk, computers have difficulty understanding and reasoning about temporal commonsense (Zhou et al., 2019).

In this paper, we focus on developing a model for temporal commonsense reasoning. Following best practices from recent work on enhancing model generalization, we propose a model that enriches pre-trained contextual representations with temporal knowledge and general commonsense knowledge by leveraging carefully selected auxiliary datasets in a multi-step fine-tuning setting. Moreover, we specifically designed a temporal task-adaptive pre-training task aimed to capture temporal commonsense knowledge by masking temporal indicators in text.

We evaluate our model on the challenging Multiple Choice Temporal Common-sense (MC-TACO) dataset (Zhou et al., 2019). In our experiments, our model substantially outperforms the standard fine-tuning approach, as well as other strong baselines.

2 Temporal Commonsense Reasoning Task

This task entirely focuses on a specific reasoning capability: temporal commonsense. Zhou et al. (2019) used crowdsourcing to create the Multiple Choice Temporal Common-sense (MC-TACO) dataset, which collects the temporal knowledge of five temporal properties: (1) duration (how long an event takes), (2) temporal ordering (typical order of events), (3) typical time (when an event occurs), (4) frequency (how often an event occurs), and (5) stationarity (whether a state is maintained for a very long time or indefinitely). It contains 13k tuples, each consisting of a sentence, a question, and a candidate answer, that should be judged as plausible or not. The sentences are taken from different sources such as news, Wikipedia and textbooks. An example from the dataset is below. The correct answer is in **bold**.

> *Paragraph*: Growing up on a farm near St. Paul, L. Mark Bailey didn't dream of becoming a judge.
> *Question*: How many years did it take for Mark to become a judge?
> a) 63 years b) 7 weeks c) **7 years**
> d) 7 seconds e) 7 hours
> *Reasoning Type*: Duration

Proceedings of the Student Research Workshop associated with RANLP-2021, pages 78–84,
held online, Sep 1–3, 2021.
https://doi.org/10.26615/issn.2603-2821.2021_012

We use the MC-TACO dataset for evaluating the performance of our model.

3 Model

Our work uses BERT (Devlin et al., 2019) as the text encoder. It has obtained high performance on several natural language understanding (NLU) benchmarks and it is relatively simple to adapt its architecture to downstream tasks. We focus on exploring different training techniques using BERT (Devlin et al., 2019), given its superior performance on a wide range of NLP tasks. The text encoder and the training methods used in our model are detailed below.

3.1 Text Encoder

BERT (Devlin et al., 2019): We use the BERT$_{\text{BASE}}$ model and the BERT$_{\text{LARGE}}$ model released by the authors. The BERT$_{\text{BASE}}$ model consists of 12 transformer layers, 12 self-attention heads per layer, and a hidden size of 768. The BERT$_{\text{LARGE}}$ model consists of 24 transformer layers, 16 self-attention heads per layer, and a hidden size of 1024.

3.2 Training Methods

Multi-Step Fine-Tuning: Multi-step fine-tuning works by performing a second stage of pre-training with data-rich related supervised tasks. It has been shown to improve model robustness and performance, especially for data-constrained scenarios (Phang et al., 2018; Camburu et al., 2019). We first fine-tune BERT on carefully selected auxiliary tasks and datasets. This model's parameters are further refined by fine-tuning on the MC-TACO dataset. The auxiliary tasks and datasets we use are detailed below:

Event Duration Prediction Task: This task involves predicting the duration of an event in a span of text. We use TimeML (Saurí et al., 2006; Pan et al., 2006), a dataset with event duration annotated as lower and upper bounds. The task is to decide whether a given event has a duration longer or shorter than a day. An example of a sentence with an event (in bold) that has a duration shorter than a day is shown below:

> In Singapore, stocks **hit** a five year low.

Event Ordering Prediction Task: This task involves predicting the temporal relationship between a pair of input events in a span of text. In our work, we use the MATRES dataset (Ning et al., 2018). It originally contains 13,577 pairs of events annotated with a temporal relation (BEFORE, AFTER, EQUAL, VAGUE). The temporal annotations are performed on 256 English documents (and 20 more for evaluation) from the TimeBank (Pustejovsky et al., 2003), AQUAINT (Graff, 2002) and Platinum (UzZaman et al., 2013) datasets. An example of a sentence with two events (in bold) that hold the BEFORE relation:

> At one point , when it **(e1:became)** clear controllers could not contact the plane, someone **(e2:said)** a prayer.

Commonsense Reasoning Task: We propose to enrich the temporal commonsense reasoning task training by leveraging data from general commonsense knowledge task. Since the commonsense reasoning task commonly also involves reasoning about temporal events, e.g. what event(s) might happen before or after the current event, we hypothesize that temporal reasoning might benefit from it. In our experiments, we use the CosmosQA (Huang et al., 2019) and the SWAG (Zellers et al., 2018) datasets. An example from the CosmosQA dataset is below. The task is to choose the correct answer among four options. The correct answer is in **bold**.

> *Paragraph*: Did some errands today. My prime objectives were to get textbooks, find computer lab, find career services, get some groceries, turn in payment plan application, and find out when KEES money kicks in. I think it acts as a refund at the end of the semester at Murray, but I would be quite happy if it would work now.
>
> *Question*: What happens after I get the refund?
>
> *Option 1*: **I can pay my bills.**
> *Option 2*: I can relax.
> *Option 3*: I can sleep.
> *Option 4*: None of the above choices.

An example from the SWAG dataset is below. The task is to choose the correct ending among four options. The correct answer is in **bold**.

> *Question*: On stage, a woman takes a seat at the piano. She
>
> *Option 1*: sits on a bench as her sister plays with the doll.

Option 2: smiles with someone as the music plays.

Option 3: is in the crowd, watching the dancers.

Option 4: **nervously sets her fingers on the keys.**

We also experimented with task-adaptive pretraining on the MC-TACO dataset followed by fine-tuning on MC-TACO. The task-adaptive pretraining method is explained below.

Task-Adaptive Pre-Training (TAPT): Although BERT achieves good performance on only fine-tuning it on the target task, there might be a distributional mismatch between the pre-trained model and the target dataset. To alleviate this issue, performing continual pre-training using the target dataset can be useful to adapt the pre-trained model to the target task (Gururangan et al., 2020). In this setting, we perform continual pre-training on BERT using the MC-TACO dataset. More specifically, we conduct the masked language modeling and the next sentence prediction tasks on BERT using the MC-TACO dataset. The masked language modeling task randomly replaces a subset of tokens by a special token (e.g., [MASK]), and asks the model to predict them. The next sentence prediction task is a binary classification task that for a given sentence pair determines whether one follows the other in the original text (Liu et al., 2020). In addition, we also experimented with masking only the tokens that have a high TF-IDF score.

Temporal Task-Adaptive Pre-Training (Temporal TAPT): In this setting, instead of randomly mask words in the masked language modeling task, we mask time-related words. Those words include numbers, adverbs, adjectives, prepositions (before/after, every, often, etc.), and units of time (hours, years, etc.).

3.3 Ensemble Model

Ensemble of deep learning models has proven effective in improving test accuracy (Allen-Zhu and Li, 2020). We built different ensemble models by taking a majority vote of the outputs of a few independently trained models. Each single model was trained on standard fine-tuning, multi-step fine-tuning, task-adaptive pre-Training, or temporal task-adaptive pre-Training using BERT.

4 Experiments

4.1 Datasets

In this paper, we use MC-TACO as the training and evaluation dataset. In addition, we use the TimeML, MATRES, CosmosQA, and SWAG datasets as auxiliary datasets in the multi-step fine-tuning setting, as detailed in Section 3.2. The summary of the datasets is shown in Table 1.

	train	val	test
MC-TACO	-	3,783	9,442
TimeML	1,248	-	1,003
MATRES	12,716	-	838
CosmosQA	25,588	3,000	7,000
SWAG	73,546	20,006	20,005

Table 1: Summary of the datasets used in our experiments.

4.2 Implementation Details

For the multi-step fine-tuning experiments, the maximum sequence length, batch size, number of epochs, and the learning rate settings are shown in Table 2. For hyperparameter tuning, the parameters with the best accuracy on performing cross-validation on the MC-TACO evaluation set are chosen. In all experiments, we use BERT$_{\text{BASE}}$ as the text encoder unless stated otherwise.

	max seq_len	batch_size	# epochs	learning rate
MC-TACO	128	{32,**16**}	{3,4,**5**}	{**1e-5**,2e-5}
TimeML	128	{32,**16**}	{3,**4**,5}	{1e-5,**2e-5**}
MATRES	128	{32,**16**}	{**3**,4,5}	{**1e-5**,2e-5}
TimeML + MATRES	128	{32,**16**}	{**3**,4,5}	{1e-5,**2e-5**}
CosmosQA	256	32	{**1**,3,5}	{1e-5,**2e-5**}
SWAG	256	32	{1,**2**,3}	{1e-5,**2e-5**}

Table 2: Hyperparameter settings for the multi-step fine-tuning experiments. The parameters with best performance are shown in **bold**.

The maximum sequence length, batch size, number of epochs, and the learning rate settings for the TAPT and Temporal TAPT experiments are set to 128, 32, 3, and 3e-5, respectively.

We use the exact match (EM) and F1-score as the evaluation metrics. EM measures how many questions a system correctly labeled all candidate answers, while F1-score measures the average overlap between one's predictions and the ground truth.

4.3 Results

Multi-Step Fine-Tuning: The results of the multi-step fine-tuning experiments are shown in Table 3. We used BERT$_{\text{BASE}}$ as the text encoder.

fine-tuned on	EM [%]	F1 [%]
MC-TACO	40.9 (42.1)	69.9 (68.2)
TimeML→MC-TACO	41.3 (40.2)	70.3 (67.1)
MATRES→MC-TACO	39.6 (42.0)	69.2 (69.4)
TimeML + MATRES →MC-TACO	40.2 (40.9)	70.2 (67.7)
CosmosQA→MC-TACO	42.2 (41.7)	70.4 (68.9)
SWAG→MC-TACO	**43.0 (42.0)**	**71.7 (67.8)**

Table 3: Test results on multi-step fine-tuning using BERT$_{\text{BASE}}$. The cross-validation results are shown in parenthesis.

The MC-TACO model denotes the model that uses standard single-stage fine-tuning using MC-TACO, and the TimeML→MC-TACO, TimeML + MATRES, the CosmosQA→MC-TACO, and the SWAG→MC-TACO models denote the models that use multi-step fine-tuning using other datasets as the first stage fine-tuning and MC-TACO as the second-stage fine-tuning. The TimeML + MATRES →MC-TACO model denotes the model that combined the TimeML and MATRES datasets for the first stage of fine-tuning. We can observe that multi-step fine-tuning improved the overall accuracy, although there were some differences depending on the dataset used. The best results were obtained when we fine-tune on SWAG followed by MC-TACO (SWAG→MC-TACO model). This indicates that enriching training with general commonsense knowledge is beneficial.

We also conducted experiments using BERT$_{\text{LARGE}}$, where we can observe similar tendency in the results compared to BERT$_{\text{BASE}}$. The results are shown in Table 4.

fine-tuned on	EM [%]	F1 [%]
MC-TACO	42.6 (42.9)	70.9 (71.0)
TimeML→MC-TACO	44.8 (43.7)	72.8 (70.8)
CosmosQA→MC-TACO	**46.3** (43.6)	73.4 (70.7)
SWAG→MC-TACO	46.2 **(44.7)**	**73.6 (72.6)**

Table 4: Test results on multi-step fine-tuning using BERT$_{\text{LARGE}}$. The cross-validation results are shown in parenthesis.

Task-Adaptive Pre-Training (TAPT): Table 5 shows the Task-Adaptive Pre-Training (TAPT) results where we randomly mask a subset of tokens. In order to check if we could further reduce the mismatch between the pre-trained model and the target task dataset, we also experimented with masking rates higher than BERT's default masking rate of 15%. However, the best accuracy was obtained with the 15% masking probability.

Masking Probability [%]	EM [%]	F1 [%]
15	**44.5 (45.2)**	**71.9 (72.4)**
30	43.5 (44.3)	71.9 (71.3)
60	42.8 (44.6)	71.1 (69.9)

Table 5: Task-Adaptive Pre-Training (TAPT) results when masking words randomly. The cross-validation results are shown in parenthesis.

We also experimented with masking the tokens that have a high TF-IDF score. For each sentence, the candidate words for masking are the top-half words with the highest TF-IDF score. We also remove the stopwords when computing the TF-IDF. We experimented with two stopwords lists: nltk stopwords list [1] and sklearn stopwords list [2]. The results are shown in Table 6. Overall, using the nltk' stopwords achieved the best results.

stopwords = nltk		
Masking Probability [%]	EM [%]	F1 [%]
15	43.3 (43.9)	71.7 (69.4)
30	43.5 (44.3)	71.7 (70.9)
45	**44.4 (42.4)**	**72.3 (69.6)**
60	43.5 (44.3)	71.7 (71.0)
stopwords = sklearn		
Masking Probability [%]	EM [%]	F1 [%]
15	42.0 (43.6)	70.9 (71.4)
30	43.0 (45.3)	71.0 (71.2)
45	**43.0 (46.0)**	**71.4 (72.5)**
60	41.7 (45.8)	70.7 (71.4)

Table 6: Task-Adaptive Pre-Training (TAPT) results when masking words with a high TF-IDF score. The cross-validation results are shown in parenthesis.

Since the accuracy differs between using nltk's stopwords and sklearn's stopwords, we looked at the contents of each stopword and found that the stopwords from sklearn contained words related to time (i.e. numbers, prepositions such as before and after, adverbs, etc.) that ended up being removed, and not being masking candidate words. Therefore, we conducted a similar experiment in which we manually removed the time-related words from the sklearn stopwords. The results of the experiment are shown in Table 7.

As we can observe, the accuracy is higher compared to when using the original sklearn's stop-

[1] https://www.nltk.org/nltk_data/

[2] https://scikit-learn.org/stable/modules/feature_extraction.html#stop-words

Masking Probability [%]	EM [%]	F1 [%]
15	42.6 (42.4)	70.6 (69.9)
30	43.2 (43.5)	71.2 (69.9)
45	**44.1 (43.8)**	**71.8 (71.3)**
60	43.3 (44.3)	71.5 (71.1)

Table 7: Task-Adaptive Pre-Training (TAPT) results when masking words with a high TF-IDF score. The cross-validation results are shown in parenthesis. Here, we use the sklearn's stopwords without the time-related words.

words, indicating that it is not optimal to exclude words related to time from the calculation of TF-IDF. The accuracy of the TF-IDF experiment is about the same as that of randomly selecting words to be masked.

Next, we set a threshold value and try to mask words where TF-IDF exceeds this value. We set different threshold values based on the percentage of the total number of words that will be masked, and we treat them as a hyperparameter. For the stopwords, we use the nltk's stopwords and the sklearn's stopwords excluding the words related to time. The results are shown in Table 8.

TF-IDF Threshold	Masking Probability [%]	EM [%]	F1 [%]
stopwords = nltk			
0.35	19.3	41.8 (42.5)	70.4 (69.2)
0.3	28.5	**42.5 (40.9)**	**71.0 (69.5)**
0.25	45.6	41.8 (42.8)	70.7 (69.0)
stopwords = sklearn - time			
0.35	22.6	42.5 (43.2)	71.2 (68.8)
0.3	31.2	42.0 (40.9)	70.3 (68.1)
0.25	43.2	**42.5 (44.6)**	**71.0 (70.3)**

Table 8: Task-Adaptive Pre-Training (TAPT) results when masking words with a high TF-IDF score. The cross-validation results are shown in parenthesis. Here, we mask all words that exceed a TF-IDF threshold value.

We found that if we masked all the words where TF-IDF exceeded the threshold, the accuracy decreases. Therefore, we randomly select tokens to mask from the words that exceed the threshold value. The results for this setting are shown in Table 9. In some cases, the accuracy was improved over the default case where the tokens are randomly masked. On the other hand, it is difficult to find regularities in the threshold setting and the percentage of masking, thus masking focusing on TF-IDF may not be effective.

Temporal Task-Adaptive Pre-Training (Temporal TAPT): Here, time-related words (numbers, before/after, every, hour, etc.) have a higher masking probability than the other words. The results are shown in Table 10. Different form the TAPT experimentes, here, masking rates higher than BERT's default masking rate of 15% improves the performance, indicating that masking temporal indicators with a higher masking rate further helps the model to acquire temporal knowledge. Moreover, we found that if we masked all the time-related words (100% masking probability), the accuracy would decrease, but if we left a few words unmasked, the accuracy improves.

TF-IDF Threshold	Masking Probability [%]	EM [%]	F1 [%]
stopwords = nltk			
0.35	15	43.0 (44.0)	71.2 (71.0)
	30	43.6 (44.6)	71.2 (70.8)
	45	43.5 (44.1)	71.3 (71.4)
	60	**44.2 (43.1)**	**72.8 (70.3)**
0.3	15	43.1 (44.9)	71.0 (71.1)
	30	43.5 (44.7)	71.6 (71.5)
	45	43.2 (44.1)	71.8 (71.2)
	60	**44.1 (44.3)**	**71.8 (71.6)**
0.25	15	43.6 (43.2)	71.4 (70.6)
	30	**43.8 (44.7)**	71.0 (72.8)
	45	42.8 (44.2)	**72.0 (70.8)**
	60	42.0 (44.7)	71.0 (70.3)
stopwords = sklearn - time			
0.35	15	42.3 (44.3)	71.3 (71.8)
	30	**45.0 (45.8)**	**72.1 (70.9)**
	45	42.9 (44.8)	70.9 (72.1)
	60	45.0 (42.6)	71.9 (68.3)
0.3	15	**44.7 (43.6)**	**72.5 (71.0)**
	30	42.6 (44.8)	71.1 (71.9)
	45	42.8 (42.5)	70.8 (70.1)
	60	43.1 (44.8)	71.3 (70.9)
0.25	15	42.9 (43.6)	71.3 (70.7)
	30	43.4 (43.8)	71.6 (71.0)
	45	42.9 (45.6)	71.3 (70.6)
	60	**43.7 (43.3)**	**71.7 (68.8)**

Table 9: Task-Adaptive Pre-Training (TAPT) results when masking words with a high TF-IDF score. The cross-validation results are shown in parenthesis. Here, we randomly select tokens to mask from the words that exceed the threshold value.

Ensemble Model: We built different ensemble models by taking a majority vote of the outputs of a few independently trained models. Each single model was trained on standard fine-tuning, multi-step fine-tuning, Task-Adaptive Pre-Training, or Temporal Task-Adaptive Pre-Training using BERT. The results are shown in Table 11.

The experimental results show that ensembling improves accuracy. In particular, *pattern 4*, which uses three models: multi-step fine-tuning with CosmosQA, multi-step fine-tuning with SWAG, and

Masking Probability (time-related words) [%]	Masking Probability (others) [%]	EM [%]	F1 [%]
100	0	42.7 (46.6)	71.0 (71.7)
90	10	44.1 (43.3)	71.6 (70.1)
80	20	**45.1 (44.3)**	**72.7 (70.7)**
70	30	42.9 (42.6)	71.9 (69.5)

Table 10: Temporal Task-Adaptive Pre-Training (Temporal TAPT) results. The cross-validation results are shown in parenthesis. Here, time-related words (numbers, before/after, every, hour, etc.) are masked with higher probability than the other words.

model	pattern1	pattern2	pattern3	pattern4
MC-TACO			✓	
TimeML →MC-TACO	✓			
CosmosQA →MC-TACO	✓	✓	✓	✓
SWAG →MC-TACO	✓	✓	✓	✓
TAPT(random)		✓		
Temporal TAPT				✓
EM [%]	45.0	45.6	44.4	**46.5**
F1 [%]	72.9	73.2	72.0	**73.9**

Table 11: Ensemble Model results.

Temporal Task-Adaptive Pre-Training obtained the best performance, with an EM score of 46.5% and an F1-score of 73.9%.

This model also outperformed the model from Zhou et al. (2019): a BERT model with standard fine-tuning, and a time unit normalized BERT model, where the authors further add unit normalization to temporal expressions in candidate answers and fine-tune on the MC-TACO dataset. Table 12 shows the results.

model	EM [%]	F1 [%]
BERT	39.6	66.1
BERT + unit normalization	42.7	69.9
Ours	**46.5**	**73.9**
Human	75.8	87.1

Table 12: Comparison of our best ensemble model with the model from Zhou et al. (2019).

We also compared our best ensemble model with TACOLM, proposed by Zhou et al. (2020). TACOLM is a BERT model pre-trained on explicit and implicit mentions of temporal commonsense, extracted from a large corpus using pattern rules. The results are shown in Table 13. As we can observe, the accuracy of all the five temporal properties was improved by our model.

class	BERT	TACOLM	Ours
Duration	33.4	34.6	**36.9**
Ordering	36.5	35.1	**46.0**
Stationarity	57.6	57.9	**59.3**
Frequency	43.3	45.1	**49.3**
Typical Time	39.5	40.9	**46.2**

Table 13: Comparison of our best ensemble model with TACOLM (Zhou et al., 2020).

4.4 Discussion

In our experiments, we could observe that multi-step fine-tuning outperforms standard single-stage fine-tuning. Also, fine-tuning BERT in the first stage using Temporal TAPT followed by fine-tuning on MC-TACO obtained the best performance among all single models. This indicates that a careful choice of the words to be masked has an impact on the performance. On the other hand, when all the words related to time were masked, the accuracy deteriorated. We hypothesize this is because if all the words were masked, the information about time would disappear from the context, and inferences about temporal common sense would be difficult.

In the TAPT experiments with TF-IDF, we found it difficult to find a regularity regarding the threshold and the ratio of masking, and it is hard to claim that masking based on TF-IDF is effective. In this study, we focus on the temporal commonsense task, and since the data contains more words related to time than other tasks, the value of IDF becomes smaller, and it may be said that TF-IDF might not be optimal.

5 Conclusion

In this paper, we proposed a model for temporal commonsense reasoning. We specifically designed a temporal masked language model task aimed to capture temporal commonsense knowledge by masking temporal indicators in text and used it in a multi-step fine-tuning setting. Moreover, we found out that an ensemble of this model with other models achieves the best results, outperforming other state-of-the-art models. In order to improve our model, we plan to conduct attention and saliency analysis.

Acknowledgements

This study was financially supported by KAKENHI (18H05521). We express our gratitude for the support.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2020. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*.

Oana-Maria Camburu, Vid Kocijan, Thomas Lukasiewicz, and Yordan Yordanov. 2019. A surprisingly robust trick for the winograd schema challenge.

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.

David Graff. 2002. *The AQUAINT corpus of English news text:[content copyright] Portions© 1998-2000 New York Times, Inc.,© 1998-2000 Associated Press, Inc.,© 1996-2000 Xinhua News Service*. Linguistic Data Consortium.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. 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 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Xiaodong Liu, Hao Cheng, Pengcheng He, Weizhu Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial training for large neural language models. *arXiv preprint arXiv:2004.08994*.

Qiang Ning, Hao Wu, and Dan Roth. 2018. A multi-axis annotation scheme for event temporal relations". In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1318–1328, Melbourne, Australia. Association for Computational Linguistics.

Feng Pan, Rutu Mulkar-Mehta, and Jerry R Hobbs. 2006. Extending timeml with typical durations of events. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 38–45.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv preprint arXiv:1811.01088*.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, Robert Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, et al. 2003. The timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*.

Roser Saurí, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. Timeml annotation guidelines. *Version*, 1(1):31.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. 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 3363–3369, Hong Kong, China. Association for Computational Linguistics.

Ben Zhou, Qiang Ning, Daniel Khashabi, and Dan Roth. 2020. Temporal common sense acquisition with minimal supervision. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7579–7589, Online. Association for Computational Linguistics.