# Salient Span Masking for Temporal Understanding

**Jeremy R. Cole** **Aditi Chaudhary** **Bhuwan Dhingra** **Partha Talukdar**
Google Research
{jrcole,aditichaud,bdhingra,partha@google.com}

## Abstract

Salient Span Masking (SSM) has shown itself to be an effective strategy to improve closed-book question answering performance. SSM extends general masked language model pretraining by creating additional unsupervised training sentences that mask a single entity or date span, thus oversampling factual information. Despite the success of this paradigm, the span types and sampling strategies are relatively arbitrary and not widely studied for other tasks. Thus, we investigate SSM from the perspective of temporal tasks, where learning a good representation of various temporal expressions is important. To that end, we introduce Temporal Span Masking (TSM) intermediate training. First, we find that SSM alone improves the downstream performance on three temporal tasks by an avg. +5.8 points. Further, we are able to achieve additional improvements (avg. +0.29 points) by adding the TSM task. These comprise the new best reported results on the targeted tasks. Our analysis suggests that the effectiveness of SSM stems from the sentences chosen in the training data rather than the mask choice: sentences with entities frequently also contain temporal expressions. Nonetheless, the additional targeted spans of TSM can still improve performance, especially in a zero-shot context.

## 1 Introduction

Salient Span Masking (SSM), first introduced by Guu et al. (2020) for retrieval-based language modeling, has shown performance gains for closed-book question answering (CBQA) (Roberts et al., 2020; Ye et al., 2020). SSM is a form of intermediate pretraining (Ye et al., 2021), where a pretrained model such as a BERT (Devlin et al., 2019) or T5 (Raffel et al., 2020) is trained further before task-specific finetuning, generally on more specialized data that does not require expensive annotations. Specifically, SSM uses the masked language modeling objective but only masks named entities and dates in sentences from English Wikipedia articles; these "salient" spans likely contain more facts, so the language model must memorize more facts in order to do the task successfully (Petroni et al., 2019). The authors use a named entity recognition model to identify entity spans and a regular expression to identify date spans. While this works well for knowledge intensive downstream tasks, such as entity-centric question answering, it remains unclear whether it is helpful for tasks that are less aligned with the data, such as common sense or temporal reasoning. Moreover, is it possible to select spans that are more related to a downstream task in order to get further performance gains?

In this work, we investigate SSM for tasks that require understanding *temporal* expressions. While SSM does include dates, the tasks we investigate include other complex temporal expressions such as durations and intervals. To that end, we introduce Temporal Span Masking (TSM): an intermediate pretraining strategy for predicting spans that are likely temporal expressions (Figure 1). Similar to SSM, TSM is automatically generated from English Wikipedia articles. We compare models trained on TSM and SSM on three temporal tasks, namely MC-TACO (Zhou et al., 2019), TimeDIAL (Qin et al., 2021) and SituatedQA (Zhang and Choi, 2021), and for one general-purpose question answering (QA) task of Natural Questions (NQ) (Kwiatkowski et al., 2019). We summarize our contributions as follows:

- We propose TSM Intermediate Training, which automatically selects temporal spans for masking.
- The new best reported results on the three temporal tasks: the best average performance is from a mixture of TSM and SSM. This mixture also does slightly better than SSM on Natural Questions.
- Experiments investigating the role of differ-



*Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3052–3060

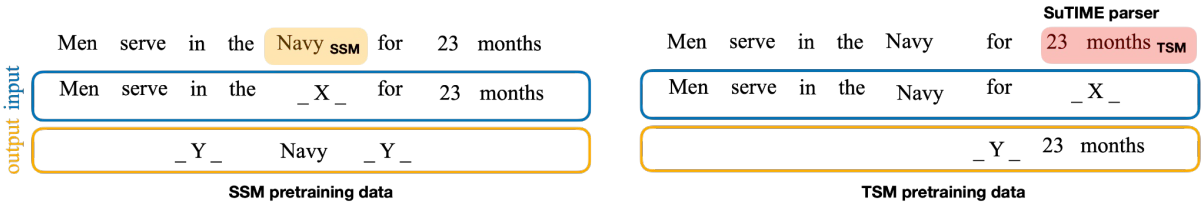


Figure 1: Overview of the TSM and SSM tasks: While SSM (Guu et al., 2020) identifies named entities and dates as salient spans, for TSM we use SUTIME parser that captures other temporal expressions such as durations and intervals. SUTIME is first applied on raw sentences to identify any temporal expressions. Next, training data for TSM is created from those sentences which have at least one temporal expression identified. In SSM, the masked spans comprise of named entity or date spans. Input to each task consists of the sentence with the selected span dropped out where the model is trained to predict the dropped tokens.

ent TSM and SSM span types, showing entity spans alone are helpful, which implies that difficult examples help improve representations of the unmasked spans as well.

## 2 Methods

Following Roberts et al. (2020), we utilize intermediate training to improve pretrained models' generalization to downstream tasks. All models are initialized from the encoder-decoder T5-1.1-XXL language model (Raffel et al., 2020), which was shown to have the best closed-book QA performance in Roberts et al. (2020).

### 2.1 Background: Salient Span Masking

Salient Span Masking (SSM) was first introduced by Guu et al. (2020) and is designed to specifically mask named entities and dates or *salient spans*. These salient spans are automatically identified from English Wikipedia using a Named Entity Recognition model to find entities as well as a regular expression to find dates. The authors mask one such span per sentence during training: the model must maximize the probability of the masked entity or date given the corrupted input sentence. Guu et al. (2020) designed the task to improve downstream performance on tasks requiring world knowledge in order to improve their retrieval-augmented model's ability to use retrieved texts. Roberts et al. (2020) then adapted this task for closed book encoder-decoder models.

### 2.2 Proposed: Temporal Span Masking

Inspired from the success of SSM, TSM is designed to address problems requiring temporal knowledge. To create training examples for TSM, we automatically identify temporal expressions in a large corpus using SUTIME (Chang and Manning, 2012), a rule-based temporal parser, that identifies temporal expressions from raw text. Given an input sentence, SUTIME is built to identify expressions of the following four types: **Time** which indicates a particular point in time such as *next Monday*, **Duration** such as *3 days*, **Set** which indicates periodic set of time that occur with some frequency such as *every 4 years* and **Date** such as *January 1*.

We run SUTIME on all of English Wikipedia[1]. Specifically, we divide the articles into sentences, and apply SUTIME[2] on each sentence. For our TSM training data, we ensure that exactly one temporal span is masked per example. So, if a sentence contains four temporal spans, we create four training examples with exactly one temporal span masked per example. Details of the temporal distribution are in Table 4. Each example is created by masking the tokens belonging to the temporal expression, as shown in Figure 1, corrupting the input sentence by replacing the span with (_X_) and having the model predict the masked tokens. The training objective is to maximize the probability of the target span given the corrupted input sentence, similar to T5's span corruption training objective (Raffel et al., 2020) and the SSM training objective (Roberts et al., 2020).

### 2.3 Model Variants

All newly reported results are based on T5-1.1-XXL models. Proposed models are named for their intermediate training objective: TSM is trained solely on the masked temporal spans described above; SSM is trained solely on the training objec-

[1] We use the 2020 snapshot of English Wikipedia (TFDS datadump *wikipedia20201201en*)

[2] https://github.com/FraBle/python-sutime

tive described in Roberts et al. (2020).[3]

We also investigate a version of the SSM pretraining data that only uses the *named entity* spans identified by the NER model; in other words, all of the *date* spans identified by the regular expression are removed, but the task is otherwise the same. We call this objective ENTITIES. Finally, we compare against models that are trained proportional mixtures of both TSM+SSM and TSM+ENTITIES.

For baseline models, we use the same pretrained model (T5-1.1-XXL) with no intermediate training (T5), as well as a T5-1.1-XXL model which has been trained for an additional 100K steps on the prefix LM task (T5-LM; Lester et al. 2021).

### 2.4 Downstream Temporal Tasks

We evaluate on three downstream temporal tasks for evaluation, finetuning a model on each task separately. Below, we briefly describe the tasks and datasets, with additional training details in Appendix A.

**MC-TACO** Zhou et al. (2019) release a human-annotated dataset to measure temporal commonsense understanding. It consists of 13k tuples of *(sentence, question, candidate answer)* covering five types of common sense problems such as event frequency, event duration, event ordering, stationarity and event typical time. Given a sentence context, a temporal question about that context, and a possible commonsense answer, the task is to determine whether the provided answer is reasonable for the given context. For instance, for the event of *taking a shower* with four possible answer choices *five minutes, fifteen minutes, fifteen hours*, and *fifteen years*, the first two are plausible and will have the *yes* label while the latter two choices would be *no*. There is no training data released for this task, so we finetune the model on the provided validation set and evaluate on the test set.

**TimeDIAL** Qin et al. (2021) release a human-annotated multi-turn dialog dataset for measuring temporal commonsense understanding in a dialog setting. The dataset comprises of challenge test set with 1.1k dialog instances derived from the Daily Dialog dataset described in Qin et al. (2021). TimeDIAL dialogs mostly comprise of common sense instances where the answers generally consist of one temporal span. For instance, in the following dialog "I'll just be a minute'., the span "a minute" may be masked out and the model is required to predict the masked span based on the dialog turns. Given a dialog with a temporal expression masked out, the task is to correctly predict which two of the four provided answers are valid in the given context. We report results without finetuning (styled TimeDIAL-0) as well as results from finetuning the model on the Daily Dialog dataset.

**SituatedQA** Zhang and Choi (2021) release an open-domain QA dataset derived from existing question answering datasets with additional annotations that resolve temporal and geographic ambiguities. Each example consists of a disambiguated question: for instance, "Which COVID-19 vaccines have been authorized in the US [as of 2020]?" or "What was the first COVID-19 vaccine to be authorized [in the US]?". For the purpose of this work, we focus on the temporal questions. These consist of 9K additional questions, with a training set of about 4.5K questions. We finetune on the training set and evaluate on the test set.

### 2.5 Natural Questions

While the focus of our method is improving temporal question answering performance, we also wanted to ensure that our method does not degrade performance on non-temporal question answering tasks. Thus, we also evaluate our model variants on Natural Questions (Kwiatkowski et al., 2019), using the "open" variant popularized by Lee et al. (2019). These examples discard those questions without short answers or that require an evidence document to answer. These consist of about 87K questions for training and an additional 3.6k questions for validation, which we use for evaluation.

## 3 Results and Discussion

Our main results can be found in Table 1. Results including Natural Questions can be found in Table 2. Note that the Natural Questions results have minor variations from published numbers; we ran these baselines ourselves, and it is possible the training setup differed slightly.

**T5 and T5-LM** The T5 model sets a relatively high baseline compared to previously reported models. The T5-LM model's extra non-domain-specific pretraining does not help on any task, suggesting

[3]We note that the SSM-spans are derived from the 2018 snapshot of English Wikipedia (same as Guu et al. (2020)) while TSM-spans from the 2020 snapshot.

| | SituatedQA | | MC-TACO | | TimeDIAL | | TimeDIAL-0 | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Model | F1 | EM | F1 | EM | 1-Best | 2-Best | 1-Best | 2-Best | Overall | Overall-0 |
| BEST REPORTED | – | 18.53 | 82.92 | 63.81 | – | 76.10 | – | 50.60 | 52.74 | 44.31 |
| T5 | 25.75 | 19.78 | 84.00 | 64.56 | 99.91 | **84.50** | 90.85 | 37.59 | 56.28 | 40.64 |
| T5-LM | 25.38 | 19.63 | 81.99 | 59.83 | 99.91 | 80.60 | 86.87 | 32.16 | 53.35 | 37.21 |
| SSM | 29.92 | 23.12 | 85.88 | 68.39 | 99.73 | 84.06 | 96.74 | 67.21 | 58.52 | 52.91 |
| ENTITIES | 29.42 | 22.82 | 85.47 | 66.59 | 99.91 | 83.06 | 97.64 | 67.93 | 57.49 | 52.45 |
| TSM | 27.42 | 21.18 | 84.89 | 65.92 | 99.91 | 83.88 | **99.82** | **77.54** | 56.99 | 54.88 |
| TSM+SSM | 29.33 | 22.76 | **86.20** | 67.64 | **100.0** | 83.78 | 98.19 | 73.10 | 58.03 | 54.5 |
| ENTITIES+TSM | **30.78** | **24.60** | 85.32 | **68.47** | 99.91 | 84.24 | 98.91 | 76.09 | **59.09** | **56.39** |

Table 1: Aggregate metrics across the three datasets. *Overall* performance is the simple arithmetic average of the harder metric for each approach (EM, EM, 2B); *Overall-0* uses TimeDIAL-0 instead of TimeDIAL. The second section contains our runs of earlier models; the Best Reported uses best known published numbers. The third section represents our models. Note that all models (in the second and third sections) are based on T5-1.1-XXL models. Best Reported results are ALBERT (Lan et al., 2019) from Abramson and Emami (2022) for TimeDIAL, BART results from Zhang and Choi (2021), and DeBERTa (He et al., 2020) results from the leaderboard for MC-TACO. Note that F1 for MC-TACO is based on the precision/recall over answers and EM is based on labeling every answer for a question correctly, while the F1 for SituatedQA is based on the token-level F1 of the answer span.

| Model | F1 | EM | Overall | Overall-0 |
|---|---|---|---|---|
| SSM | 41.57 | 34.6 | 52.54 | 48.33 |
| T5 | 39.35 | 32.38 | 50.31 | 38.58 |
| T5-LM | 37.16 | 31.14 | 47.80 | 35.69 |
| ENTITIES | 41.21 | 34.52 | 51.75 | 47.97 |
| TSM | 39.24 | 32.69 | 50.92 | 49.33 |
| TSM+SSM | 41.80 | 35.10 | 52.3 | 49.65 |
| ENTITIES+TSM | **41.89** | **35.18** | **53.11** | **51.09** |

Table 2: Results on Natural Questions – *Overall* and *Overall-0* results include the same metrics from Table 1 with Natural Questions (EM) included. The first section represents our baselines. Note that all models are based on T5-1.1-XXL models.

extra training steps does not in of itself cause improvements on these tasks.

**Entities** The ENTITIES model, which is trained on only non-temporal entity spans, performs better overall than the TSM task. It only does worse on the TimeDIAL dataset, which is almost entirely focused on conversational, non-knowledge based contexts. It still does substantially better than the base T5 model when no finetuning data is available. This high performance is possibly due to the prevalence of temporal spans in the SSM training data. Running SUTIME on the ENTITIES data reveals that 45% of its training examples contain at least one date, duration, set, or time. This suggests that sentences with named entities in general already carry temporal-salient information useful for downstream temporal tasks. See Appendix B for a full breakdown of the co-occurrences.

**SSM** The SSM model is the second best overall. It benefits from both its own date spans as well as the frequent presence of temporal spans in the entities data, suggesting difficult example sentences are more important than the type of masked span. It does worse on TimeDIAL-0, however, where the task is to score the best temporal span.

**TSM** The TSM model improves upon the baseline T5 model but is worse overall than the SSM model. However, it is the best on TimeDIAL-0. This is likely because the DailyDialog training dataset is relatively large, which may overcome the need for intermediate pretraining altogether. Note that TSM achieves a mild performance improvement over the baseline T5 model on Natural Questions, but is notably worse than the other intermediate training methods.

**TSM+SSM** The TSM+SSM model improves over TSM but is worse than SSM outside of TimeDIAL-0. One possible reason for the regression is that TSM and SSM have overlapping Date span examples, which may make the intermediate task easier and thus less useful. However, it is slightly better than SSM on Natural Questions.

**Entities+TSM** The ENTITIES+TSM model performs the best overall: with and without the extra training data for TimeDIAL. It has the benefit of

| Model | T5 | SSM | E+TSM |
|---|---|---|---|
| Duration | 88.64 | 88.28 | **89.09** |
| Set | 86.39 | **88.22** | 87.79 |
| Time | 87.74 | **88.70** | 87.70 |
| Date | 65.75 | 66.63 | **66.87** |
| Entities | 42.17 | 45.76 | **46.54** |

Table 3: Aggregated performance across types for baselines and the best overall model (ENTITIES+TSM). For TimeDIAL, each answer span is labeled by SUTIME. Duration includes MC-TACO's "Event Duration"; Set includes MC-TACO's "Frequency", Time and Date both include MC-TACO's "Typical Time"; Date also includes SituatedQA; Entities include all of MC-TACO and SituatedQA. Note that the majority of the gains from SSM/TSM seem to be from Entities and Dates. See Appendix B for rationale behind these choices.

TSM spans without containing overlapping spans or losing the world knowledge from entity spans. It also performs slightly better than SSM on Natural Questions.

**By Type** We analyze model performance by temporal type in Table 3. The main improvement of both SSM and ENTITIES+TSM is in entity and date tasks. Surprisingly, TSM shows a regression on time tasks, and only gets a slight improvement on duration tasks. One possible hypothesis for this is that temporal expressions may be more informative when co-occurring with an entity. Note that these numbers are based on the trained versions of each dataset, excluding Natural Questions. Note that SituatedQA contains further breakdowns based on the scope of the date, but this does not map well to the other datasets.

## 4 Related Work

**Span Masking and Intermediate Training** Salient Span Masking (Guu et al., 2020) came out of a series of efforts like SpanBERT (Joshi et al., 2020) to select more difficult examples to improve models memorization of the text.

Most similar to us, Ye et al. (2021) explore a similar paradigm of choosing better spans for a downstream task (e.g., entity linking or relation extraction) where they experiment with both a heuristic masking policy similar to SSM and also a learned masking policy. They similarly find that masking spans that resemble downstream tasks improve performance, however, they also note that learned masking policies suffer from overfitting. Yang et al. (2020) and Zhou et al. (2020) explore intermediate training by designing heuristics to identify sentences containing temporal expressions and then adding additional tasks and losses, rather than using span masking. TSM differs in more closely resembling the pretraining task.

Levine et al. (2021) use pointwise mutual information to jointly mask highly correlated spans to avoid the model relying on local signals but rather learning from the broad context. They find this leads to faster and better pretraining. In the future, it might be interesting to see how PMI-spans can combine with knowledge-oriented span techniques such as SSM, TSM, and whether they can help in the intermediate training paradigm.

**Temporal Understanding** There has been a surge of interest in probing models' temporal awareness. While we evaluate on a three tasks, it is far from an exhaustive evaluation and we leave further evaluations of our method to future work.

Recently, Thukral et al. (2021) and Vashishtha et al. (2020) construct NLI datasets to test whether pretrained models understand certain types of common sense temporal expressions, such as containment. To probe common sense, we use TimeDIAL (Qin et al., 2021) for its naturalistic dialogues as well as MC-TACO (Zhou et al., 2020), which uses a diverse set of situations and temporal expressions.

For factual questions, open-response temporal questions are closely aligned with our work (e.g., TimeQA; Chen et al. 2021; TempLAMA; Dhingra et al. 2022). All of TempLAMA, TimeQA, and SituatedQA (Zhang and Choi, 2021) rely primarily on the year as the main temporal expression being tested, where facts are scoped to the provided years. To probe temporally scoped facts, we use SituatedQA for its more naturalistic questions.

## 5 Conclusion

In this work, we investigate SSM as it relates to temporal tasks that require understanding both commonsense and world knowledge questions and propose a new intermediate training method which selects spans generated by a temporal parser. These intermediate training strategies result in the best overall reported results on the selected downstream tasks. However, we find that even the entity spans from SSM are helpful for temporal tasks, likely because entity-containing examples also contain informative temporal knowledge.

## Limitations

This analysis investigates only the encoder-decoder model architecture: in particular, encoder-only models such as BERT (Devlin et al., 2019) and decoder-only models such as GPT-2 (Radford et al., 2019) are excluded. Further, large language models, such as PaLM (Chowdhery et al., 2022) or GPT-3 (Brown et al., 2020) are also not investigated. See Appendix B for further discussion.

## Acknowledgements

We would like to thank Kelvin Guu and Srini Narayanan, as well as our anonymous reviewers, for their helpful feedback on a previous version of this manuscript.

## References

Darren Abramson and Ali Emami. 2022. An application of pseudo-log-likelihoods to natural language scoring. *arXiv preprint arXiv:2201.09377*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Angel X. Chang and Christopher Manning. 2012. SUTime: A library for recognizing and normalizing time expressions. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3735–3740, Istanbul, Turkey. European Language Resources Association (ELRA).

Wenhu Chen, Xinyi Wang, and William Yang Wang. 2021. A dataset for answering time-sensitive questions. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

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.

Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

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

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yoav Levine, Barak Lenz, Opher Lieber, Omri Abend, Kevin Leyton-Brown, Moshe Tennenholtz, and Yoav Shoham. 2021. {PMI}-masking: Principled masking of correlated spans. In *International Conference on Learning Representations*.

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.

Lianhui Qin, Aditya Gupta, Shyam Upadhyay, Luheng He, Yejin Choi, and Manaal Faruqui. 2021. TIMEDIAL: Temporal commonsense reasoning in dialog. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7066–7076, Online. 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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Shivin Thukral, Kunal Kukreja, and Christian Kavouras. 2021. Probing language models for understanding of temporal expressions. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 396–406, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Siddharth Vashishtha, Adam Poliak, Yash Kumar Lal, Benjamin Van Durme, and Aaron Steven White. 2020. Temporal reasoning in natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4070–4078, Online. Association for Computational Linguistics.

Zonglin Yang, Xinya Du, Alexander Rush, and Claire Cardie. 2020. Improving event duration prediction via time-aware pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3370–3378, Online. Association for Computational Linguistics.

Qinyuan Ye, Belinda Z Li, Sinong Wang, Benjamin Bolte, Hao Ma, Wen-tau Yih, Xiang Ren, and Madian Khabsa. 2020. Studying strategically: Learning to mask for closed-book qa. *arXiv preprint arXiv:2012.15856*.

Qinyuan Ye, Belinda Z. Li, Sinong Wang, Benjamin Bolte, Hao Ma, Wen-tau Yih, Xiang Ren, and Madian Khabsa. 2021. On the influence of masking policies in intermediate pre-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7190–7202, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Michael Zhang and Eunsol Choi. 2021. SituatedQA: Incorporating extra-linguistic contexts into QA. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7371–7387, Online and Punta Cana, Dominican Republic. 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.

## A Training Details

All models are initialized from the public T5-1.1-XXL checkpoints.[4]

### A.1 Intermediate Training

We use 256 Cloud TPU v3 cores for the intermediate training procedure using a batch size of 2048 and the fixed default learning rate of 0.001. Training generally proceeds for one epoch, which is between 100-150K steps depending on the precise task, though we used early stopping for the TSM and TSM+SSM models based on MC-TACO performance, as they seem to overfit.

### A.2 Finetuning

We use 64 Cloud TPU v3 cores for finetuning and inference on all tasks. For MC-TACO, Natural Questions, and SituatedQA, we use the same fixed learning rate of 0.001 and train for 10K steps with batch size 128. For TimeDIAL, we attempt to follow their training setup more closely, and use a lower learning rate (0.0001) and train for up to 100K steps, still with batch size 128 and use early stopping on the validation set to inform when to stop. For most of the models that had intermediate training, the early stopping point was for 10K steps. However, for the basic T5 model, it was after 20K steps (improving from $(82.97 \rightarrow 84.51)$), implying that it can overcome its lack of intermediate training with additional finetuning data. Note that in the zero-shot variant, no finetuning is done.

## B Further Discussion

### B.1 Other Experiments

We previously experimented with the T5 Large and T5-XL models, as well the as 1.0 versions of the T5 models that were first described in Raffel et al. (2020). In general, larger models and the 1.1 versions worked better. While we refrain from reporting results due to inconsistent setups, in general the smaller models were notably worse, such that distinguishing between two similar setups (such as TSM and SSM) was difficult on many tasks. While we know of no work testing salient span masking on extremely large models, it is possible it would actually show a larger impact, based on this trend. While left-to-right decoding serves as an awkward fit for the paradigm, if our hypothesis on the reason why SSM works is correct, then it should not prove to be a substantial hurdle. See also below for more discussion on said hypothesis.

[4] https://github.com/google-research/text-to-text-transfer-transformer

| Temporal Type | Number of sentences |
|---|---|
| Date | 56,520,912 |
| Duration | 8,182,819 |
| Set | 1,797,929 |
| Time | 2,281,198 |

Table 4: TSM data statistics: The above table describes the distribution of temporal spans in the English Wikipedia data, which comprises of 121M sentences.

| Span Type | Number of sentences |
|---|---|
| Entity | 78,139,341 |
| Date | 32,023,769 |

Table 5: SSM data statistics: The above table describes the distribution of salient spans in the Wikipedia data as processed by (Guu et al., 2020), which comprises of 82M sentences. Each row denotes the number of sentences that contain at least one of the respective span.

| TSM Span Type | SSM Span Type | |
|---|---|---|
| | Named Entity | Date |
| Date | 29,771,242 | - |
| Duration | 5,159,229 | 3,144,592 |
| Set | 1,226,333 | 844,222 |
| Time | 915,084 | 411,436 |

Table 6: We apply SUTIME on the SSM training data (Guu et al., 2020) to investigate how many sentences contain temporal information. Each column denotes the number of sentences that contain the SSM identified span (e.g. *named entity* or *date*) and each row denotes the number of those sentences in which SUTIME identified the corresponding temporal span. Number of sentences with at least one *named entity*: 78,139,341 Number of sentences with at least one *date*: 32,023,769 (Table 5)

### B.2 Span Distribution vs. Text Distribution

Our hypothesis for SSM's effectiveness is due to it oversampling difficult sentences. This is based on the performance gain for the ENTITIES intermediate training as well as the number of temporal spans that occur in the SSM training data. Table 6 shows the results of SUTIME parser on the SSM training data, and as we can see, significant portion

of the SSM data (45%) has temporal spans. Table 6 shows the breakdown of different temporal spans for each SSM salient span type. We leave an exact test of this for future work, but if this is true, then we might expect left-to-right decoding models to also benefit from the sampling procedure of SSM, even though they do not use a masked language modeling paradigm for training.

### B.3 Span Types in Table 4

Mapping MC-TACO's span types is somewhat helpful to see the performance breakdown. Note that these are now based on individual answers, while MC-TACO's strict match metric is based on correctly labeling all answers for a given question.

**Entities** While MC-TACO is a common sense dataset, it frequently relies on reasoning about relatively complicated phenomena. While it is common sense to know that a dynasty does not rule in China for only a few minutes, it is still required to know more about China and dynasties to answer the question correctly. TimeDIAL on the other hand is normally ordinary conversations that are not very entity-centric. SituatedQA is derived from Natural Questions, which is an information seeking dataset that frequently features entities.

**Duration** MC-TACO's event duration maps well to the Duration type in SUTIME. While there may be some SituatedQA examples that include durations, we do not filter for them.

**Set** MC-TACO's Frequency type asks question of the "How often" nature while sets frequently have answer types of that nature e.g., "every third sunday", but this is not a perfect mapping.

**Date** MC-TACO's typical time sometimes includes dates, but it is less likely to be a specific date and more likely to be a generic date like Sunday, rather than a specific knowledge-based date. SituatedQA questions always include dates that decontextualize Natural Questions.

**Time** MC-TACO's typical time sometimes corresponds with times as well, but they are again less likely to be specific. Unfortunately, Date and Time are not separated in MC-TACO.

**Other MC-TACO Types** Note that we did not include the "Stationarity" or the "Event Ordering" MC-TACO types in the breakdown, as they do not correspond well to any SUTIME type.