*Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022)*, pages 4451–4457
Marseille, 20-25 June 2022


# Improving Event Duration Question Answering by Leveraging Existing Temporal Information Extraction Data

**Felix Giovanni Virgo, Fei Cheng, Sadao Kurohashi**
Graduate School of Informatics, Kyoto University
Kyoto, Japan
{felix,feicheng, kuro}@nlp.ist.i.kyoto-u.ac.jp

**Abstract**

Understanding event duration is essential for understanding natural language. However, the amount of training data for tasks like duration question answering, i.e., McTACO, is very limited, suggesting a need for external duration information to improve this task. The duration information can be obtained from existing temporal information extraction tasks, such as UDS-T and TimeBank, where more duration data is available. A straightforward two-stage fine-tuning approach might be less likely to succeed given the discrepancy between the target duration question answering task and the intermediary duration classification task. This paper resolves this discrepancy by automatically recasting an existing event duration classification task from UDS-T to a question answering task similar to the target McTACO. We investigate the transferability of duration information by comparing whether the original UDS-T *duration classification* or the recast UDS-T *duration question answering* can be transferred to the target task. Our proposed model achieves a 13% Exact Match score improvement over the baseline on the McTACO duration question answering task, showing that the two-stage fine-tuning approach succeeds when the discrepancy between the target and intermediary tasks are resolved.

**Keywords:** event duration, temporal common sense, question answering, data recasting

## 1. Introduction

Understanding how long an event typically lasts is essential in natural language processing. Many NLP tasks, such as narrative understanding, event timeline construction, question answering, and natural language inference (Nakhimovsky, 1987; Ning et al., 2018; Zhou et al., 2019; Leeuwenberg and Moens, 2019; Vashishtha et al., 2020), require knowledge about the typical duration of events. However, it is still challenging for machines to comprehend the duration of various events available. An event verb can have different durations depending on its context. For example, "*take a vacation*" takes longer than "*take a shower*." While taking a shower typically takes a few minutes, a vacation can last for days or even weeks. Acquiring the duration of various events by hand is also costly and time-consuming.

McTACO (Zhou et al., 2019) is a temporal commonsense question answering dataset that consists of questions from 5 temporal phenomena including event duration. The event duration questions are the focus of our paper. This work shows that the performance of modern pre-trained NLP models for this task is still far behind humans. Since the amount of training data only covers a limited number of events and their attributes, incorporating external event duration information is necessary to improve this task.

Leveraging relevant intermediary tasks has shown to be beneficial for improving target tasks with limited data (Phang et al., 2018; Liu et al., 2019). For example, Liu et al. (2019) shows that it is beneficial to fine-tune the target RTE (Bentivogli et al., 2009) task starting from the intermediary MultiNLI (Williams et al., 2018) model. RTE is a binary entailment task similar to MultiNLI, but with much less training data. In our case, the target task is a duration question answering task, such as the event duration problems from McTACO. The external duration information can be obtained from an existing temporal information extraction task, such as UDS-T (Vashishtha et al., 2019) or TimeBank (Pan et al., 2011).

In temporal commonsense question answering, given a context, a time-related question, and a list of candidate answers, the task is to find the plausible answers from the list of candidate answers. It is possible for a question to have multiple plausible answers. Consider the following example from McTACO:

> ***Context***: *Mohamed Atta was born on September 1, 1968, in Kafr el Sheikh, Egypt, to a middle-class family headed by his father, an attorney.*
> ***Question***: *How many years did Atta live with his parents?*
> ***Answer 1***: *18 years.*
> ***Answer 2***: *20 years.*
> ***Answer 3***: *18 months.*

The event being asked "*live*" is not explicitly stated in the context. Still, we can infer that children usually live with their parents until they become adults, which makes the plausible answers are "18 years" and "20 years." Since the event being asked might or might not be explicitly stated in the context, we need to encode a tuple of (*context*, *question*, *one candidate answer*) into a single sentence and get its sentence-level representa-

tion to predict whether it is *plausible* or *not plausible*. However, this is different from the duration classification task, where the task is to predict the duration unit of each event in the context. Consider the following example from UDS-T:

> *Their worker even **cleaned** 3 of my windows and **changed** a lightbulb.*

The event "*cleaned*" in this context usually lasts for *minutes* or *hours*. To predict the duration unit of an individual event "*cleaned*," we need to explicitly get its event-level representation instead of the sentence-level representation since multiple events could exist in a single context sentence. The target duration question answering task requires *implicit* event encoding, whereas the intermediary duration classification task requires *explicit* event encoding. Given the discrepancy between the target and intermediary tasks, a straightforward two-stage fine-tuning approach might be less likely to succeed.

In this paper, we aim to improve the performance of the target McTACO duration question answering task by resolving its discrepancy with the intermediary UDS-T duration classification task. We propose a novel method to recast an existing event duration classification task from UDS-T to automatically construct a new duration question answering dataset similar to McTACO. We investigate the transferability of duration information from the recast UDS-T data to the target McTACO by experimenting with two-stage fine-tuning on pre-trained language models with recast UDS-T data as the intermediary. Our proposed model outperforms the baseline RoBERTa model by 13% on the Exact Match score on the McTACO duration question answering task, suggesting that the two-stage fine-tuning approach succeeds when the discrepancy between the target and intermediary tasks are resolved.

## 2. Related Work

**Duration Question Answering.** Zhou et al. (2019) annotate a temporal commonsense question answering dataset called McTACO that consists of questions from 5 temporal phenomena: *duration*, *ordering*, *typical time*, *frequency*, and *stationarity*. They use this dataset to probe the capability of various systems, such as BERT (Devlin et al., 2019), on temporal commonsense understanding. Zhou et al. (2020) present a transformer-based temporal common sense language model called TACOLM, trained on temporal data that are extracted using patterns from a large corpus. It uses temporal data from 3 temporal commonsense dimensions: *duration*, *frequency*, and *typical time*. It outperforms BERT in various temporal tasks, including McTACO.

**Event Duration Datasets.** Pan et al. (2011) annotate the events in TimeBank (Pustejovsky et al., 2003) with their expected durations by specifying upper and lower bounds. They compare different learning algorithms to classify the events into two class based on their duration: *less than a day* and *a day or longer*. Gusev et al. (2011) use web query patterns in an unsupervised approach to predict the typical duration of various events. Vashishtha et al. (2019) annotate a temporal relation and duration dataset called UDS-T. They propose a joint method that extracts both temporal relations and event durations between a pair of events.

## 3. Duration Data Recasting

We recast an existing temporal dataset, UDS-T, which contains annotations for event duration to construct a new event duration question answering dataset, UDST-DurationQA[1]. UDS-T is annotated on top of the Universal Dependencies English Web Treebank (Silveira et al., 2014), and it consists of 32k events and 70k event-event relations. For each event in an event pair, the annotation contains the *start point* and *end point* of the event in the timeline (starting from 0 to 100), alongside its duration unit. We choose UDS-T as the source of external duration information to improve the McTACO duration question answering task given its relatively large size. Figure 1 shows the example of the recast UDST-DurationQA from the original UDS-T dataset.

**Step 1. Irrelevant Contexts Removal.** We first remove some of these texts from English Web Treebank that might not suit our target task. There are five genres in the corpus: *weblogs*, *newsgroups*, *email*, *reviews*, and *question-answers*. We remove texts from two genres: *weblogs* and *email*. *Weblogs* contains news articles with discussions, while *Email* contains emails sent by employees of a company. Context sentences from these genres mostly are replies to the discussions, making it hard to understand the bigger topics on their own. We also remove contexts that are too short (less than 10 words) or too long (more than 36 words) since they are usually just short utterances that do not have much meaning or they contain too many different ideas in a sentence.

**Step 2. Question Generation.** We use AlllenNLP[2]'s Semantic Role Labeling model (Shi and Lin, 2019) to extract the semantic roles related to an event in a sentence, i.e the subject and the object of the event. For each event, we formulate the question as: *How long does it take for* `[subject]` *to* `[event]` `[object]`*?* If the subject is a subjective pronoun, it is transformed into its objective pronoun, e.g., from "he" to "him." The event verb is transformed into its lemma using `LemmInflect`[3], e.g., from "went" to "go."

**Step 3. Candidate Answer Generation.** We generate 6 to 8 candidate answers for each question, consisting of 2 to 3 positive answers and 4 to 5 negative answers, around the same number as McTACO. We formulate

[1]Our recast dataset is available at `https://github.com/felixgiov/UDST-DurationQA`

[2]`https://github.com/allenai/allennlp`

[3]`https://github.com/bjascob/LemmInflect`

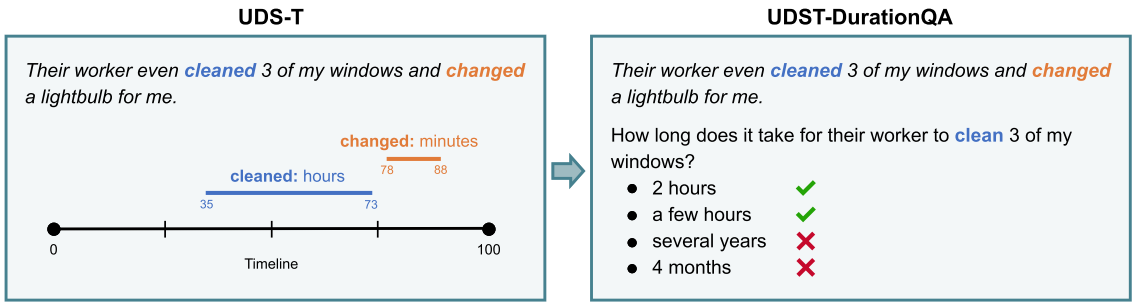


Figure 1: Example of the recast UDST-DurationQA from the original UDS-T dataset.

the candidate answer as: `[number] [duration unit]`, e.g., "30 minutes" and "2 weeks." For the `[duration unit]`, we use the duration labels in UDS-T, such as: *seconds*, *minutes*, *hours*, *days*, *weeks*, *months*, *years*, *decades*, and *centuries*. For other UDS-T labels, i.e., *instantaneous* and *forever*, since they are not duration units, we do not use `[number]` in it. Instead, we use phrases like "it takes instantly" and "it takes forever" as the candidate answers. Positive answers and negative answers differ on how we generate the `[number]` and the choice of the `[duration unit]`.

To generate **positive answers**, we rely on the duration spans and units of the events in the event pair. A duration span of an event is defined as the difference between its *end point* and its *start point*. Given a pair of events, we define $e_1$ as the event with the longer duration span and $e_2$ as the event with the shorter span. For both $e_1$ and $e_2$, we randomly generate a `[number]` between the lower bound and the upper bound of their respective duration unit. For example, *hours* has lower bound of 1 and upper bound of 24. To generate the `[number]` of $e_1$ with a more precise range, we compute the new upper bound for $e_1$ relative to the span of $e_2$, as shown in Equation 1. For example, consider we know $e_1$ lasting *hours*, $e_2$ lasting *minutes*, and the span of $e_1$ is 4 times longer than $e_2$. Assuming $e_2$ lasts at most 60 minutes (upper bound of *minutes*), $e_1$ should last at most 240 minutes or 4 hours, which becomes the new upper bound for $e_1$. We also apply the same logic to compute the new lower bound for $e_2$, as shown in Equation 2.

$$upper_{e_1} = \frac{span_{e_1}}{span_{e_2}} \times upper_{e_2} \quad (1)$$

$$lower_{e_2} = \frac{span_{e_2}}{span_{e_1}} \times lower_{e_1} \quad (2)$$

We randomly perturb the candidate answer by replacing `[number]` with a determiner word, such as "a few" or "several," in 1 out of 4 occurrences. This way, we can generate candidate answers more similar to human-annotated answers in McTACO, where not all of them contain the exact number of the duration.

For both $e_1$ and $e_2$, we use their respective duration unit as the `[duration unit]`.

For each of the **negative answers**, we randomly select the `[duration unit]` where it is at least two units apart from the positive answers. If the positive answer is in *hours* then the negative answers cannot be in *minutes* or *days*. We choose two units apart since the adjacent temporal units are also likely to be the temporal units of an event via *approximate agreement* (Pan et al., 2011). We randomly generate the `[number]` between the normal lower bound and the upper bound of the duration unit, without considering the relation to other events.

**Statistics.** Table 1 shows the number of unique question-answer pairs in McTACO-duration and UDST-DurationQA for each split. McTACO-duration is a subset of McTACO whose questions are about event duration. UDST-DurationQA uses the same split as the original UDS-T dataset. Table 2 shows the number of unique questions for each split in McTACO-duration and UDST-DurationQA. Based on the number of unique questions, assuming one question is asking about one event, the number of events in UDST-DurationQA is around 16 times larger than McTACO-duration. The small number of events in McTACO implies the lack of training data, which indicates the need for external duration data. Figure 2 shows the duration distribution of positive answers in UDST-DurationQA and McTACO-duration. In UDST-DurationQA, there is a relatively high number of events lasting *minutes*, with a relatively even distribution across other duration units. This distribution is still relatively similar to the original UDS-T distribution. Meanwhile, in McTACO-duration, events lasting *years* are the ones with the highest number, followed by *minutes* and *hours*. There are only few events lasting *decades* or more in both datasets.

## 4. Two-stage Fine-tuning Approach

We fine-tune a pre-trained language model, such as BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019) to perform the duration question answering task and duration classification task. These models were trained on Masked Language Model task and large text

| Dataset | Train + Dev | Test |
|---|---|---|
| McTACO-duration | 1,112 | 3,032 |
| UDST-DurationQA | 40,103 + 4,924 | 4,868 |

Table 1: Number of unique question-answer pairs in each dataset. UDST-DurationQA uses the same split as the original UDS-T dataset.

| Dataset | # questions |
|---|---|
| McTACO-duration | 439 |
| UDST-DurationQA | 7,082 |

Table 2: Number of unique question for each dataset.

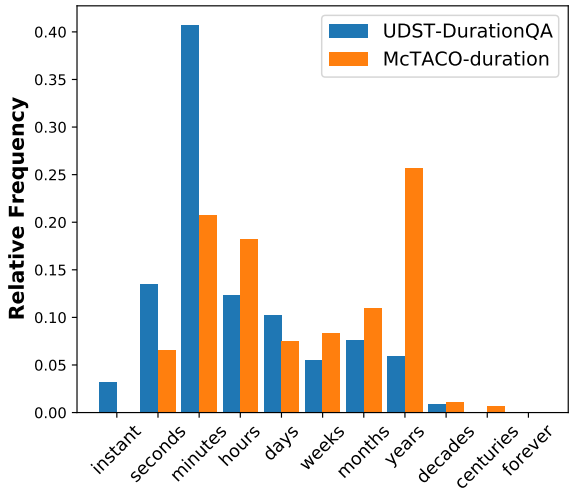


Figure 2: Duration distribution of positive answers in UDST-DurationQA and McTACO-duration.

data using bidirectional Transformer (Vaswani et al., 2017), enabling contextualized representation that can be used to fine-tune a broad range of downstream tasks. Figure 3 shows our two-stage fine-tuning approach. First, we fine-tune a pre-trained language model on an intermediary task, i.e., duration question answering or duration classification tasks. In the next stage, we fine-tune the model on the McTACO duration question answering task.

**Duration Question Answering Model.** We formulate the duration question answering task as a binary sequence-pair classification task. The model receives two elements: (1) the context sentence concatenated with the question and (2) a candidate answer, separated with a special token (`[SEP]` token in BERT). The final hidden state of the first token in the sequence (`[CLS]` token in BERT) is fed into a dense output layer to make a binary prediction on each instance, *plausible* or *not plausible*.

**Duration Classification Model.** We formulate the duration classification task as a multi-class token classification task. The model receives a context sentence and the positions of each event in the sentence. The final hidden state of the token corresponding to each event is fed into a dense output layer to predict the duration label of each event in the context sentence. For example, in UDS-T duration classification task, there are 11 duration labels: *instantaneous*, *seconds*, *minutes*, *hours*, *days*, *weeks*, *months*, *years*, *decades*, *centuries*, and *forever*.

## 5. Experiments

### 5.1. Model Implementation and Evaluation Metrics

We use the `transformers`[4] (Wolf et al., 2020) library from HuggingFace to implement our model. We use a batch size of 16 with Adam (Kingma and Ba, 2015) as the optimizer. For UDS-T duration classification task and UDST-DurationQA, we use an initial learning rate of 1e-5 and train the models for 2 epochs. For McTACO-duration, we use an initial learning rate of 2e-5 and train the models for 10 epochs.

Same as McTACO, we use two different metrics to evaluate the model performance: (1) Exact Match (EM), which measures how many questions a system is able to correctly label all candidate answers, and (2) F1, which measures the average overlap between predictions and the ground truth.

### 5.2. Experimental Settings

To investigate how well our UDST-DurationQA dataset can benefit McTACO on duration questions, we compare 4 models with the following settings:

1. **Baseline model.** RoBERTa-large model that is fine-tuned only on McTACO-duration.

2. **Baseline model.** RoBERTa-large model that is fine-tuned on UDS-T duration classification task then fine-tuned on McTACO-duration.

3. **Proposed model.** RoBERTa-large model that is fine-tuned on a variant of UDST-DurationQA, whose candidate answers only consist of `[duration unit]` without the `[number]`, then fine-tuned on McTACO-duration. This setting can be directly compared to the Setting 2 since this setting's task and UDS-T duration classification task are both duration unit prediction tasks without numbers involved.

4. **Proposed model.** RoBERTa-large model that is fine-tuned on UDST-DurationQA then fine-tuned on McTACO-duration.

### 5.3. Results and Discussion

Table 3 shows the Exact Match and F1 scores of 4 different model settings on McTACO-duration. In line

[4] `https://github.com/huggingface/transformers`

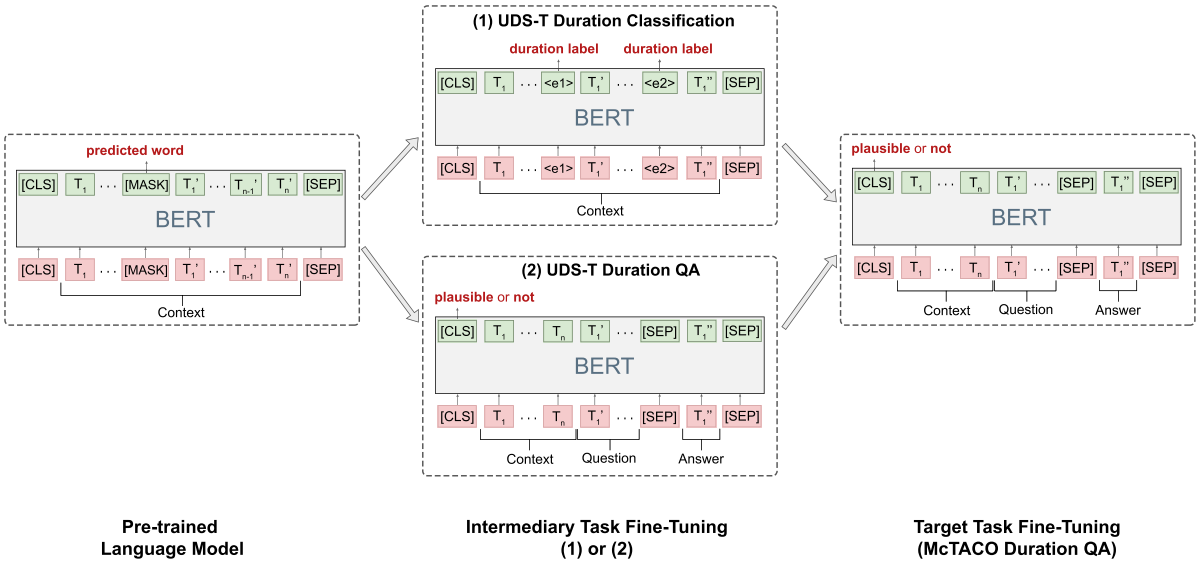


Figure 3: Two-stage fine-tuning.

| No | Model | EM | F1 |
|---|---|---|---|
| 1 | $\text{RoBERTa}_{\text{large}} \rightarrow$ McTACO-duration | 40.45 | 67.42 |
| 2 | $\text{RoBERTa}_{\text{large}} \rightarrow$ UDS-T (duration classification) $\rightarrow$ McTACO-duration | 39.49 | 64.95 |
| 3 | $\text{RoBERTa}_{\text{large}} \rightarrow$ UDST-DurationQA (unit only) $\rightarrow$ McTACO-duration | 42.78 | 66.97 |
| 4 | $\text{RoBERTa}_{\text{large}} \rightarrow$ UDST-DurationQA $\rightarrow$ McTACO-duration | **45.86** | **70.52** |

Table 3: Performances on McTACO-duration between different experimental settings. The scores are the average of 3 runs with different random initializations. Arrow indicates the fine-tuning process. All scores are in percentages. Higher is better.

with our hypothesis on the discrepancy issue, we observe degradation in the performance when we fine-tune RoBERTa-large on UDS-T duration classification task as the intermediary compared to the baseline RoBERTa-large model without any intermediary task. On the other hand, fine-tuning RoBERTa-large on UDST-DurationQA as the intermediary improves the Exact Match score by 5.4 points (around 13%) and F1 score by 3.1 points (around 5%) compared to the baseline RoBERTa-large model. These results show that leveraging an intermediary duration task succeeds in improving the target duration task when the discrepancy between the target and intermediary tasks are resolved. Additionally, adding the numbers in the UDST-DurationQA candidate answers improves the Exact Match score by 3.1 points and F1 score by 3.5 points compared to the one that has duration unit only. This indicates the importance of the duration number in the target task. This also shows that the model can differentiate small and large numbers in the duration, albeit the same duration unit.

By comparing the predictions of our proposed model and the baseline model, we observe the most significant improvement on events lasting *seconds*, followed by *days* and *minutes*. The improvement generally corresponds to the distribution of UDST-DurationQA, where the events lasting from *seconds* to *days* are the ones with the highest numbers. These events usually exist in both UDST-DurationQA and McTACO-duration and share the same domains, e.g., *prepare food* or *enter a building*. On the other hand, the models struggle to improve events that last a long time (more than 10 years), e.g., *form a fossil* which can take tens of thousands of years. Besides the lack of data in these domains, we think this could also happen because answers in McTACO tend to use *years* as the unit to describe this type of event, e.g., *50 years* or *1,000 years*, as opposed to *decades* and *centuries* in UDST-DurationQA.

## 6. Additional Experiments

### 6.1. Comparison with Pre-trained Temporal Language Model

We also compare our proposed method to a pre-trained temporal common sense language model TACOLM (Zhou et al., 2020). TACOLM is a transformer-based language model trained on temporal signals from 3 temporal commonsense dimensions, including *duration*, that are acquired with minimal supervision from a large corpus. To ensure a fair comparison, we use BERT-base (Devlin et al., 2019) in-

| Model | EM | F1 |
|---|---|---|
| TACOLM (Zhou et al., 2020) → McTACO | 34.60† | - |
| $\text{BERT}_{\text{base}}$ → McTACO | 33.76 | 60.98 |
| $\text{BERT}_{\text{base}}$ → UDST-DurationQA → McTACO | **36.52** | **63.22** |

Table 4: Performances on McTACO-duration between TACOLM, BERT, and our model. All models are fine-tuned on all of McTACO data and not just the duration questions. Our scores are the average of 3 runs with different random initializations. † indicates the reported Exact Match score from the paper (F1 score is not available).

| Model | EM | F1 |
|---|---|---|
| Two-stage Fine-tuning | **45.86** | **70.52** |
| Multi-task Learning | 41.72 | 66.93 |

Table 5: Performances on McTACO-duration between two-stage fine-tuning and multi-task learning with UDST-DurationQA. The scores are the average of 3 runs with different random initializations.

stead of RoBERTa-large since BERT is the baseline model being compared in the TACOLM paper. Same as TACOLM, we also fine-tune the model on all of McTACO data and not just the duration questions. Table 4 shows the performances on McTACO-duration between TACOLM and our proposed method. Fine-tuning TACOLM on McTACO achieves a higher Exact Match score than fine-tuning on the BERT-base model. Our proposed method outperforms both TACOLM and BERT-base models on Exact Match by 1.9 and 2.7 points, respectively.

### 6.2. UDST-DurationQA Performance

We also evaluate the performance of UDST-DurationQA task. We fine-tune RoBERTa-large on UDST-DurationQA train set and evaluate the model on the test set. Model implementation and hyperparameters are the same as in Section 5.1. The number of question-answer pairs used for training and testing is shown in Table 1. The model achieves an Exact Match score of 40.12 and an F1 score of 72.49. While it is not exactly comparable to the McTACO scores, we think this is a reasonable performance given that UDST-DurationQA is automatically created with different data sizes and domains compared to McTACO.

### 6.3. Multi-task Learning

We experiment with multi-task learning to investigate whether setups that leverage two-stage fine-tuning are more effective than multi-task learning. In multi-task learning, we jointly fine-tune RoBERTa-large on both intermediary and target tasks. We empirically weight the loss of McTACO-duration to 0.9 and UDST-DurationQA to 0.1. This is to avoid the bias towards UDST-DurationQA since the McTACO is our main task, and the number of UDST-DurationQA is much larger than McTACO. We train the model for 10 epochs. Table 5 shows the Exact Match and F1 scores of the two-stage fine-tuning approach compared to the multi-task learning approach on McTACO-duration test set. Naive multi-task learning yields worse performance scores than two-stage fine-tuning. We think two-stage fine-tuning better suits our case because the number of data for the intermediary task is much larger than the target task.

## 7. Conclusion

In this paper, we recast an existing temporal information extraction dataset, UDS-T, to construct a new event duration question answering dataset similar to the target McTACO, with the aim to resolve the discrepancy between the two different duration tasks. We experiment with fine-tuning recast UDS-T data as the intermediary before the target McTACO data to investigate the transferability of duration information between these two datasets. Our proposed model outperforms several baseline pre-trained models on the McTACO duration question answering task. We also present our recast dataset as a new resource for the duration question answering task to contribute to future research in temporal common sense.

## Acknowledgements

We thank the reviewers for their helpful feedback. This work was supported by the project KAKENHI: 21H00308.

## References

Bentivogli, L., Magnini, B., Dagan, I., Dang, H. T., and Giampiccolo, D. (2009). The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the Second Text Analysis Conference, TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009*. NIST.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (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, June. Association for Computational Linguistics.

Gusev, A., Chambers, N., Khilnani, D. R., Khaitan, P., Bethard, S., and Jurafsky, D. (2011). Using query patterns to learn the duration of events. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Yoshua Bengio et al., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Leeuwenberg, A. and Moens, M.-F. (2019). A survey on temporal reasoning for temporal information extraction from text. *Journal of Artificial Intelligence Research*, 66:341–380, 09.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv e-prints*, page arXiv:1907.11692, July.

Nakhimovsky, A. (1987). Temporal reasoning in natural language understanding: The temporal structure of the narrative. In *Proceedings of the Third Conference on European Chapter of the Association for Computational Linguistics*, EACL '87, page 262–269, USA. Association for Computational Linguistics.

Ning, Q., Wu, H., and Roth, D. (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, July. Association for Computational Linguistics.

Pan, F., Mulkar-Mehta, R., and Hobbs, J. R. (2011). Annotating and learning event durations in text. *Computational Linguistics*, 37(4):727–752, December.

Phang, J., Févry, T., and Bowman, S. R. (2018). Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *CoRR*, abs/1811.01088.

Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., and Lazo, M. (2003). The timebank corpus. *Proceedings of Corpus Linguistics*, 01.

Shi, P. and Lin, J. (2019). Simple bert models for relation extraction and semantic role labeling. *ArXiv*, abs/1904.05255.

Silveira, N., Dozat, T., de Marneffe, M.-C., Bowman, S., Connor, M., Bauer, J., and Manning, C. (2014). A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Vashishtha, S., Van Durme, B., and White, A. S. (2019). Fine-grained temporal relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2906–2919, Florence, Italy, July. Association for Computational Linguistics.

Vashishtha, S., Poliak, A., Lal, Y. K., Van Durme, B., and White, A. S. (2020). Temporal reasoning in natural language inference. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4070–4078, Online, November. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In I. Guyon, et al., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana, June. Association for Computational Linguistics.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October. Association for Computational Linguistics.

Zhou, B., Khashabi, D., Ning, Q., and Roth, D. (2019). "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *EMNLP*.

Zhou, B., Ning, Q., Khashabi, D., and Roth, D. (2020). Temporal common sense acquisition with minimal supervision. In *ACL*.