Temporal Validity Change Prediction

Anonymous ACL submission

Abstract

Temporal validity is an important property of text that has many downstream applications, such as recommender systems, conversational AI, and user status tracking. Existing benchmarking tasks often require models to identify the temporal validity duration of a single statement. However, many data sources contain additional context, such as successive sentences in a story or posts on a social media profile. This context may alter the duration for which the originally collected statement is expected to be valid. We propose *Temporal Validity Change Prediction*, a natural language processing task benchmarking the capability of machine learning models to detect context statements that induce such change. We create a dataset consisting of temporal target statements sourced from Twitter and crowdsource corresponding context statements. We then benchmark a set of transformer-based language models on our dataset. Finally, we experiment with a multi-tasking approach to improve the state-of-the-art performance.

1 Introduction

Information is not impervious to time. Whether it be a post on a social media timeline like "I am going grocery shopping", a statement like "Barack Obama is the president of the United States" in a knowledge repository, or an advertisement like "Ariana Grande concert in town this weekend", sentences frequently contain inherently time-sensitive information. Consequently, readers have to reason over whether the statement is still current and accurate when they ingest the information. This property of a statement can be described as its *temporal validity* (Almquist and Jatowt, 2019; Hosokawa et al., 2023; Lynden et al., 2023). Similar to previous work in the growing field of temporal commonsense reasoning (Wenzel and Jatowt, 2023; Jain et al., 2023), determining the temporal validity of a statement often relies on our prior commonsense understanding of the world due to underspecified temporal expressions. However, it can be difficult to accurately reason over this property when sentences actually belong to a larger content stream. For example, extracting a single post from a social media timeline or a single sentence from a book may remove important contextual information about the statement's validity period.

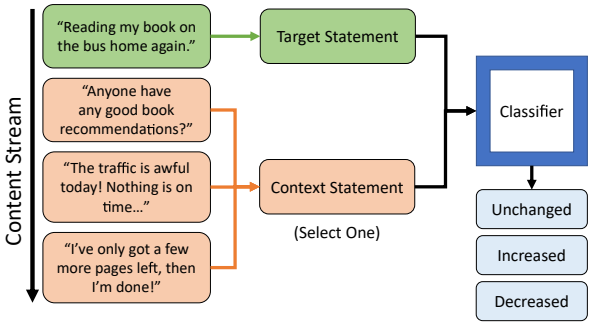

Figure 1: A visualization of the TVCP task. A context statement from the content stream (e.g., a timeline or news article) provides additional information about the temporal validity duration of the target statement.

We follow the previous work by Hosokawa et al. (2023), who pioneer the idea of evaluating context statements that explicitly support or contradict the temporal validity of a target statement. However, we instead focus on identifying context statements that impact the *duration* for which a statement may be valid. The resulting task can be used to benchmark a different type of temporal reasoning in machine learning models, but could also be used to provide an additional classification signal for existing duration-based temporal validity classifiers (Almquist and Jatowt, 2019; Lynden et al., 2023). We propose a new natural language processing task called *Temporal Validity Change Prediction* (TVCP) to model this problem, which is visualized in Figure 1. Some example applications for this task are listed below.

Timeline Prioritization: Social media services

such as Twitter rely on recommender systems to prioritize the vast amount of content that their users produce. One possible way to improve the prioritization of content is to consider its temporal validity (Takemura and Tajima, 2012; Koul et al., 2022), as users are likely to be more interested in current and relevant statements over outdated ones. TVCP can be used to leverage the stream of social media posts by a given user as possible context to better estimate the temporal validity duration of any previously observed post.

User Status Tracking: Similarly, the content of a user's posts on social media could be utilized for other analytical or business purposes, such as predicting revenue streams (Asur and Huberman, 2010; Deng et al., 2011; Lassen et al., 2014; Lu et al., 2014) or identifying trends in a community's or an individual user's behaviour (Li et al., 2018; Abe et al., 2018; Shen et al., 2020). TVCP could be used to identify posts that refer to previous time-sensitive information, to detect chains of thought about topics that may not be self-contained.

Conversational AI: Foundation models, such as CHATGPT (Ouyang et al., 2022) and BARD (Manyika, 2023), could incorporate the temporal validity of statements provided by the user to keep track of knowledge that is still relevant to the conversation. Using TVCP, new messages could be used as context to adjust the expected temporal validity period of the previously learned facts. This is especially relevant as recent reports indicate that foundation models may struggle with temporal commonsense reasoning (Bian et al., 2023; Jain et al., 2023).

Our main contributions are the following:

1. We define a novel natural language processing task titled *Temporal Validity Change Prediction*, which requires models to predict the impact of a context statement on a target statement's temporal validity duration.

2. We build a dataset composed of time-sensitive *target statements*, as well as *follow-up statements* that provide context.

3. We evaluate the performance of a set of transformer-based *language models* (LMs) on our dataset, including *large language models* (LLMs) such as CHATGPT (Ouyang et al., 2022), GPT-4 (Achiam et al., 2023), LLAMA2 (Touvron et al., 2023), and MIXTRAL (Jiang et al., 2024).

4. We propose an augmentation to the fine-tuning process that leverages temporal validity duration labels to improve the performance of the state-of-the-art classifier.

2 Related Work

2.1 Temporal Commonsense Reasoning

Temporal commonsense reasoning is considered one of several categories of commonsense reasoning (Storks et al., 2019; Bhargava and Ng, 2022). A major driver of research specifically into temporal common sense appears to have been the transformer architecture (Vaswani et al., 2017) and resulting LMs. In recent years, several datasets that specifically aim to benchmark temporal commonsense understanding have been published (Zhou et al., 2019; Ning et al., 2020; Zhang et al., 2020; Qin et al., 2021; Zhou et al., 2021), while ROC-STORIES (Mostafazadeh et al., 2016) appears to be the only dataset focussing on this type of reasoning before the publication of the transformer architecture. Small adjustments to transformer-based LMs are often proposed as state-of-the-art solutions for these datasets (Pereira et al., 2020; Yang et al., 2020; Zhou et al., 2020; Pereira et al., 2021; Kimura et al., 2021; Zhou et al., 2021, 2022; Cai et al., 2022; Yu et al., 2022). Similarly, temporalized transformer models are popular solutions for tasks such as document dating or semantic change detection (Rosin and Radinsky, 2022; Rosin et al., 2022; Wang et al., 2023).

The temporal commonsense taxonomy defined by Zhou et al. (2019) is frequently referenced. It contains the five dimensions of *duration* (how long an event takes), *temporal ordering* (typical order of events), *typical time* (when an event happens), *frequency* (how often an event occurs) and *stationarity* (whether a state holds for a very long time or indefinitely).

2.2 Temporal Validity

Compared to temporal commonsense reasoning, temporal validity of text is a less well-researched field. It effectively combines three dimensions of the taxonomy by Zhou et al. (2019): *Stationarity*, to reason about whether a statement contains time-sensitive information, *typical time*, to reason about when the time-sensitive information occurs, and *duration*, to reason about how long the time-sensitive information takes to resolve.

Method	Task	Data Source	Duration Bias	Model	# Samples
Takemura and Tajima (2012)	TVDP	Twitter	N/A	SVC	9,890
Almquist and Jatowt (2019)	TVDP	Blogs, News, Wikipedia	*years*	SVC	1,762
Hosokawa et al. (2023)	TVR	Image Captions	*seconds*[1]	LM	10,659
Lynden et al. (2023)	TVDP	WikiHow	*hours*	LM	339,184
Ours	TVCP	Twitter	*hours*	LM	5,055

Table 1: Summary of related work

Takemura and Tajima (2012) classify the lifetime duration of tweets, i.e., the informational value of a tweet over time. They use handcrafted, domain-specific features to train a *support vector classifier* (SVC) on supervised samples. Similarly, Almquist and Jatowt (2019) design features to estimate the temporal validity duration of sentences collected from news, blog posts, and Wikipedia into one of five possible classes, also using SVCs. Hosokawa et al. (2023) define the *Temporal Natural Language Inference* (TNLI) task. The goal is to determine whether the temporal validity of a given hypothesis sentence is supported by a premise sentence. Lynden et al. (2023) build a large dataset of human annotations specifying the duration required to perform various actions on WikiHow as well as their respective temporal validity durations.

2.3 Comparison with Related Work

Table 1 shows the most closely related research. As noted, our dataset is based on the proposed TVCP task, whereas the previous work was based on tasks that we denote as *Temporal Validity Duration Prediction* (TVDP) and *Temporal Validity Reassessment* (TVR), the latter being our name for the TNLI task. These three tasks are described in more detail in Section 3.

Another prominent distinctive attribute is the text source and the resulting temporal validity duration bias. For example, sentences sourced from news or Wikipedia articles often appear to be valid for years or longer. On the other hand, image captions may only contain ongoing information for a few seconds or minutes. We decided to source our sentences from Twitter due to its alignment with our downstream use cases. Similar to Lynden et al. (2023), our collected time-sensitive information tends to be valid for a few hours.

We follow recent research by evaluating our dataset using transformer-based LMs, whereas earlier approaches relied on methods such as SVCs.

Except for the CoTAK dataset (Lynden et al., 2023), the datasets tend to be relatively small. As crowdsourcing is used in all datasets referenced in Table 1 to annotate text spans with commonsense information, the costs of dataset creation can quickly escalate. In addition, we ask participants to create sample context statements. This approach further restricts the overall size of our dataset due to the relative difficulty of the task.

[1]Based on analysis of a sample. TVDP labels are not available for the full dataset.

3 Task

3.1 Defining Temporal Validity

As shown in Equation 1, the temporal validity of a statement s at a time t is a binary value that determines whether the information in s is valid (true) at the given time.

$$\mathrm{TV}(s,t) = \begin{cases} \text{True} & \text{if information in } s \text{ is valid at } t, \\ \text{False} & \text{otherwise} \end{cases} \quad (1)$$

Note that some statements, known as *stationary statements*, do not contain any time-sensitive information (e.g., "Japan lies in Asia"). As expected, the temporal validity of such a statement is constant for any timestamp t. For the purposes of TVCP, we ignore these statements, as context is unlikely to change their validity. For statements containing time-sensitive information, we assume a statement is valid from the moment of sentence conception until the information is no longer ongoing.

3.2 Formalizing Existing Tasks

Temporal Validity Duration Prediction (TVDP) TVDP is the primary task that is evaluated in temporal validity research (Takemura and Tajima, 2012; Almquist and Jatowt, 2019; Lynden et al., 2023). The goal is to estimate the duration for which a statement is valid, starting at the statement creation time. We formalize this task in Equation 2, where t_s is the timestamp at which the statement s is created.

$$\mathrm{TVDP}(s) = \max_{t \geq t_s} \{t \mid \mathrm{TV}(s,t) = \text{True}\} \quad (2)$$

The TVDP task is useful in domains such as social media, where information on the posting

time of a statement is readily available and can be used to infer the timespan during which the statement is valid.

Temporal Validity Reassessment (TVR) TVR, defined also as *Temporal Natural Language Inference* by Hosokawa et al. (2023), is a task whose purpose is to infer whether a *target statement* (s_t) is temporally valid, given additional context in the form of a *follow-up statement* (s_f). The goal of the task is a reassessment of the temporal validity of s_t, that is, whether s_t is still temporally valid at t_{s_f}, given the information in s_f. Formally, we define TVR in Equation 3 (SUO = supported, INV = invalidated, UNK = unknown), where $\text{TV}^c(s,t)$ is the temporal validity of a statement s at a time t given context c. The UNK class is assigned in cases where $\text{TV}^{s_f}(s_t, t_{s_f})$ is neither clearly supported nor invalidated by the context.

$$\text{TVR}(s_t, s_f) = \begin{cases} \text{SUO} & \text{TV}^{s_f}(s_t, t_{s_f}) = \text{True} \\ \text{INV} & \text{TV}^{s_f}(s_t, t_{s_f}) = \text{False} \\ \text{UNK} & \text{TV}^{s_f}(s_t, t_{s_f}) = \text{Unclear} \end{cases} \quad (3)$$

Unlike TVDP, this task format does not require an explicit temporal anchoring of the target statement to reason over its validity, making it particularly useful for downstream applications such as story understanding, wherein a larger text stream of individual statements is provided with no clear temporal anchoring of statements. We propose *Temporal Validity Reassessment* as a new name for this task moving forward, to scope and align it with other tasks in the temporal validity domain.

3.3 Temporal Validity Change Prediction

In the context of the two tasks described above, we propose *Temporal Validity Change Prediction* (TVCP). Similar to TVR, we require s_t and s_f as an input for classification, and determine a ternary label that provides information about the impact of s_f on s_t. However, while TVR can be considered a standalone temporal validity reasoning process, TVCP primarily provides an additional signal for estimating the temporal validity duration of a statement, and is best used in conjunction with TVDP and a downstream task format where explicit temporal anchors for target statements can be derived. On top of acting as a new reasoning benchmarking task, a model trained on the TVCP task can also evaluate possible context statements for a given target statement, which could help bootstrap a larger-scale dataset creation process for future research into contextual temporal validity estimation. Formally, we define TVCP in Equation 4 (DEC = decreased, UNC = unchanged, INC = increased), where $\text{TVDP}^c(s)$ is the temporal validity duration of a statement s given context c. Figure 2 shows a concrete comparative example of all three tasks.

$$\text{TVCP}(s_t, s_f) = \begin{cases} \text{DEC} & \text{TVDP}(s_t) > \text{TVDP}^{s_f}(s_t) \\ \text{UNC} & \text{TVDP}(s_t) = \text{TVDP}^{s_f}(s_t) \\ \text{INC} & \text{TVDP}(s_t) < \text{TVDP}^{s_f}(s_t) \end{cases} \quad (4)$$

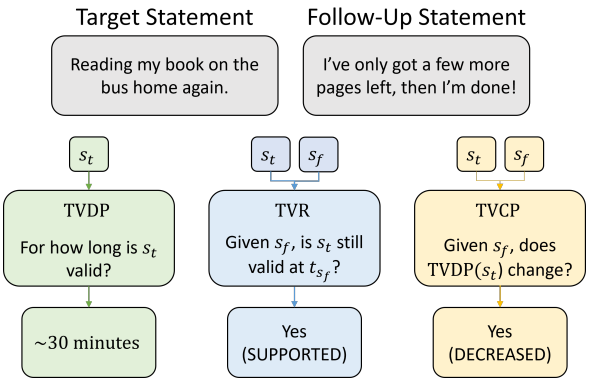

Figure 2: An example of TVDP, TVR and TVCP. Even when a context statement *supports* that an action is still ongoing (TVR), that same context may *decrease* the expected temporal validity duration (TVCP).

Of note are the implicit semantic roles of s_t and s_f. While s_f acts as additional contextual information, any information that is newly introduced in s_f should not be evaluated on its temporal validity. Our goal is exclusively to estimate the change in the temporal validity duration of s_t.

We find that temporal validity change generally occurs along two dimensions. The first dimension is *implicit* versus *explicit* change. For example, an appointment mentioned in a target statement may be declared postponed in the follow-up statement, which would be an explicit change. On the other hand, the author may instead note in a follow-up statement that the appointment is for a surgery, which may cause us to re-evaluate the duration of the appointment, although the information itself has not changed. The second dimension is a change to the *occurrence time* versus the *duration* of the information. For example, a flight may be delayed, in which case the occurrence time changes. Alternatively, the flight might have to be re-routed mid-air due to bad weather, in which case the dura-

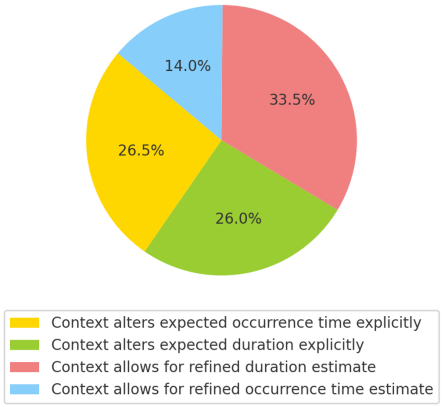

Figure 3: Causes of temporal validity duration change and frequency in our qualitative analysis

tion changes. Figure 3 shows the frequency of each combination of dimensions in a qualitative analysis of 200 context statements from our dataset. Generally, changes to the duration tend to be slightly more frequent than changes to the occurrence time. This makes sense, as context is unlikely to change the occurrence time of already ongoing information, which constitutes a large part of our target statements.

4 Dataset

We create a dataset for training and benchmarking TVCP, where each sample is a quintuple $< s_t, s_f, \text{TVDP}(s_t), \text{TVDP}^{s_f}(s_t), \text{TVCP}(s_t, s_f) >$.

s_t consists of posts sampled from Twitter. We apply several preprocessing steps to minimize unwanted statements (e.g., those containing spam, offensive content, external context, or stationary information). The collection pipeline is explained in more detail in Appendix A. Our code, including all preprocessing steps, is published under the Apache 2.0 licence.

We note that in some previously created datasets (Hosokawa et al., 2023; Lynden et al., 2023), the scope of evaluated time-sensitive information is limited to actions, such as "I am *baking bread*". However, we show that other types of time-sensitive information exist, such as events (e.g., in the sentence "*Job interview* tomorrow") or temporary states (e.g., in the sentence "It *is nice out* today"). In a qualitative analysis of 100 target statements, shown in Figure 4, we find that these alternative types of time-sensitive information constitute a significant portion (28%) of samples. Additionally, one-third of sampled statements contain at least two distinct pieces of time-sensitive information with differing temporal validity spans. This indicates that our dataset may be more diverse with respect to the richness of the evaluated time-sensitive information, compared to previous work.

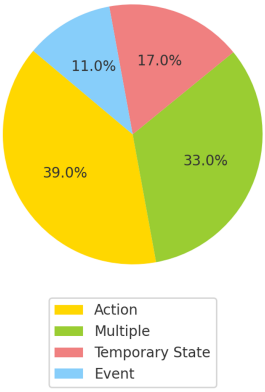

Figure 4: Distribution of different types of time-sensitive information in our qualitative analysis

For each target statement s_t, we ask two crowdworkers to estimate $\text{TVDP}(s_t)$ from the logarithmic class design shown in Equation 5, which is modelled after human timeline understanding (Jatowt and Au Yeung, 2011; Varshney and Sun, 2013; Howard, 2018). If the annotators disagreed, we supplied a third vote. We discarded any tweets that were annotated as *less than one minute* or *more than one month*, tweets that annotators tagged as being stationary, and tweets where no majority agreement could be reached, meaning our dataset is solely composed of temporal statements that achieved a majority vote label. Of 2,996 annotated target tweets, 571 were discarded without a third annotation, 867 were added without a third annotation, 546 were discarded after providing a third vote, and 1,012 were added after providing a third vote. The resulting label distribution is shown in Figure 5.

$$t \in \{< 1 \text{ minute}, 1\text{-}5 \text{ minutes}, 5\text{-}15 \text{ minutes}, 15\text{-}45 \text{ minutes}, 45 \text{ minutes-}2 \text{ hours}, 2\text{-}6 \text{ hours}, \text{more than } 6 \text{ hours}, 1\text{-}3 \text{ days}, 3\text{-}7 \text{ days}, 1\text{-}4 \text{ weeks}, \text{more than } 1 \text{ month}\} \quad (5)$$

Both s_f and $\text{TVDP}^{s_f}(s_t)$ were provided by a separate set of crowdworkers, given s_t and $\text{TVDP}(s_t)$ as an input. The updated temporal validity duration label is provided by the same participant that provides s_f. This means the target TVCP label, which is derived from the duration labels, is guaranteed to match author intent.

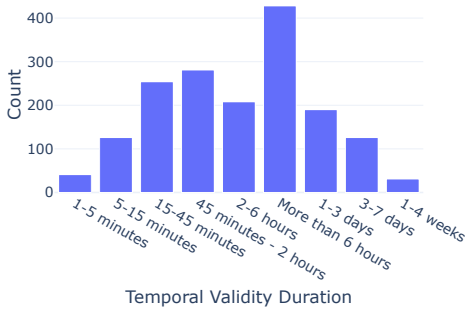

Figure 5: Distribution of TVDP labels (before temporal validity change) in our dataset

In Figure 6, we plot the *temporal validity change delta*, which is the class distance between the original and the updated TVDP estimate. We find that, in most cases, the temporal validity duration of a target statement is shifted only by one class.

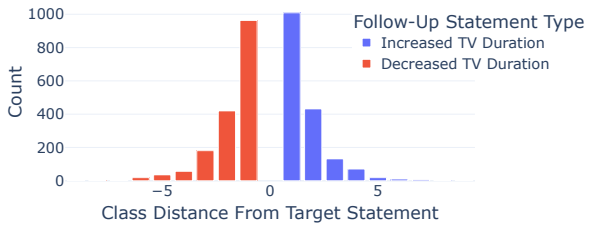

Figure 6: Temporal validity change delta distribution

For each target statement, we generate three samples covering all TVCP classes (thus ensuring a balanced dataset), each with distinct s_f, $\text{TVDP}^{s_f}(s_t)$, and $\text{TVCP}(s_t, s_f)$ labels. We collected 5,055 samples from 1,685 target statements, noting average word counts for s_t (16.1, $\sigma = 4.40$) and s_f (14.6, $\sigma = 3.29$). Similar to the TNLI dataset, our crowdsourced context statements have similar length to the target statements, but with lower variance, likely due to crowdworkers aiming to replicate the form of provided statements. Further, our collected target statements are longer on average and have a higher variance. This makes sense, as image captions, the source of TNLI samples, are likely to be shorter and more similar in length compared to randomly sampled tweets.

Crowdsourcing tasks were set up on Amazon Mechanical Turk, using qualification tests, participation criteria, and manual verification of results to ensure high-quality samples. The concrete task setup for both crowdsourcing tasks is described in Appendix B. Other than the annotations, the only data we collect is sample counts and work times by each pseudonymized participant. We do not collect any personal or identifying data. We publish the resulting dataset for public use under the CC BY 4.0 licence. In accordance with the Twitter developer policy[2], we only publish the Tweet IDs of sourced statements.

5 Experiments

5.1 Language Models

The evaluated models include fine-tuned transformer-based LMs as well as LLMs prompted in a few-shot setting. We evaluate BERT and RoBERTa as baseline models. SelfExplain (Sun et al., 2020), which achieved state-of-the-art results on the TNLI dataset, and still performs very competitively on datasets such as SST-5 (Socher et al., 2013) and SNLI (Bowman et al., 2015), represents a state-of-the-art transformer-based classification model. We initialize these models with pre-trained weights and fine-tune them on our dataset. For LLMs, we evaluate GPT-3.5-turbo, GPT-4-turbo, Mixtral8x7B, and Llama 2 in a few-shot setting. Our prompt is based on chain-of-thought reasoning (Wei et al., 2022). Further, we follow the TELeR taxonomy (Santu and Feng, 2023) to the best of our abilities to create an appropriate prompt. Our prompt is single-turn and instruction-based with a defined system role, and contains a high-level general directive, bullet-list style subtasks, few-shot samples, and an explicit statement asking the LLM to explain its own output. The models, their parameter counts, and the evaluation types are listed in Table 2. The training and prompting process is described in more detail in Appendix C.

Model Name	# Params	Evaluation
BERT-base	110M	Fine-tuned
RoBERTa-base	125M	Fine-tuned
SelfExplain	127M	Fine-tuned
Mixtral-8x7B	47B	Few-shot
Llama 2	70B	Few-shot
GPT-3.5-turbo	N/A	Few-shot
GPT-4-turbo	N/A	Few-shot

Table 2: A summary of evaluated models

For all fine-tuned models, we also provide a *multitask implementation*, in which we add two regression layers that aim to respectively predict $\text{TVDP}(s_t)$ and $\text{TVDP}^{s_f}(s_t)$ from the same hidden representation. For these layers, we calculate the mean squared error between a single output neuron and a linear mapping of the TVDP class

[2] https://developer.twitter.com/en/developer-terms/policy, accessed 12.10.2023

index to the range $[0, 1]$. Our intuition is that embeddings with an understanding of TVDP labels may be better suited for TVCP. Inspiration for this approach are models that utilize the interplay between temporal dimensions to improve their temporal commonsense reasoning performance, such as SYMTIME (Zhou et al., 2021) or SLEER (Cai et al., 2022). The number of trainable parameters added by this approach is negligible.

5.2 Evaluation

Evaluation Metrics

We evaluate two metrics, accuracy and *exact match* (EM). Accuracy is simply the fraction of correctly classified samples. EM is the fraction of *target statements* for which all three corresponding samples were correctly classified. This metric punishes inconsistency in the model more strictly, thus providing a better view of the true performance and task understanding of each model (Wenzel and Jatowt, 2023), while disincentivizing shallow reasoning behaviours commonly seen in transformer models (Helwe et al., 2021; Tan et al., 2023).

We report the mean EM and accuracy across a five-fold cross-validation split. Each split consists of 70% training data, 10% validation data, and 20% test data. The results are shown in Table 3. In the remainder of this section, we refer to the best-performing model, SELFEXPLAIN with multitask fine-tuning, as MULTITASK.

Model	$\overline{\text{Acc}}$ (+ MT)	$\overline{\text{EM}}$ (+ MT)
LLAMA 2	46.5 (N/A)	9.7 (N/A)
MIXTRAL-8X7B	63.0 (N/A)	22.5 (N/A)
GPT-4-TURBO-1106	69.3 (N/A)	30.4 (N/A)
GPT-3.5-TURBO-1106	67.9 (N/A)	31.1 (N/A)
ROBERTA	78.7 (+1.1)	48.2 (+2.1)
BERT	84.8 (−0.2)	61.2 (+0.9)
SELFEXPLAIN	**88.5 (+ 1.1)**	**69.8 (+ 2.8)**

Table 3: Model evaluation results, sorted by mean EM score. MT = Multitask Implementation.

Foundation Model Performance

In our evaluation, few-shot prompted foundation models consistently rank far below fine-tuned, smaller LMs, including simple baselines such as BERT and ROBERTA. This is consistent with previous research (Bian et al., 2023; Jain et al., 2023), which shows that temporal reasoning is an area in which foundation models are lacking. However, the few-shot learning approach most definitely leads to a lack of knowledge about dataset specifics traits that a trained classifier could leverage, which partially explains the discrepancy.

LLAMA 2 in particular suffers from a high rate (26.07%) of explanations that violate the prompt by not providing one of the three target classes. This behaviour is not seen in other LLMs. Interestingly, while accuracy slightly increases for GPT-4-TURBO compared to GPT-3.5-TURBO, the EM score does not increase, meaning there is no need to resort to exceedingly large models to achieve state-of-the-art few-shot performance on TVCP, but also raising questions over why the temporal reasoning in GPT-4-TURBO stagnates compared to its smaller counterparts in this instance.

When breaking the classification accuracy down by the temporal validity change delta (Figure 7), MULTITASK strongly outperforms GPT-3.5-TURBO on both neutral and non-neutral context statements. However, while the performance of MULTITASK is relatively stable, GPT-3.5-TURBO performance decreases when the context causes small changes to the temporal validity duration.

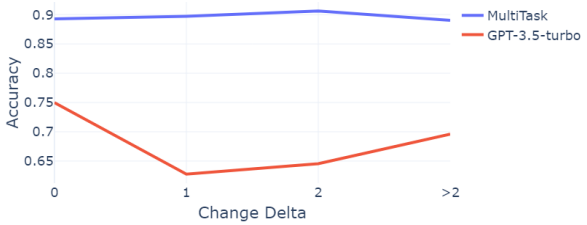

Figure 7: Temporal validity change delta vs. accuracy in MULTITASK and GPT-3.5-TURBO

Multitasking

We note a positive impact on the EM score from implementing multitasking in all fine-tuned models. To measure the statistical significance of implementing multitask learning on SELFEXPLAIN, we use a technique known as bootstrapping. We resample from both classification sets 10,000 times with replacement and evaluate the fraction of resulting samples where MULTITASK outperforms SELFEXPLAIN. We find $p = 0.0012$ for accuracy, with a 95% confidence interval of $[0.0036, 0.0192]$. For EM, the significance is smaller at $p = 0.0216$, with a 95% confidence interval of $[0.0006, 0.0397]$, as the number of samples is smaller due to being based on the number of target statements. Additionally, to evaluate the impact of training data quantity on classifier performance, we train the MULTITASK classifier on a single train-val-test split (80%/10%/10%) with different amounts of training data. The results can be seen in Figure 8.

Here, the model does not yet appear to be saturated.

Figure 8: Training data quantity vs. performance metrics in MULTITASK

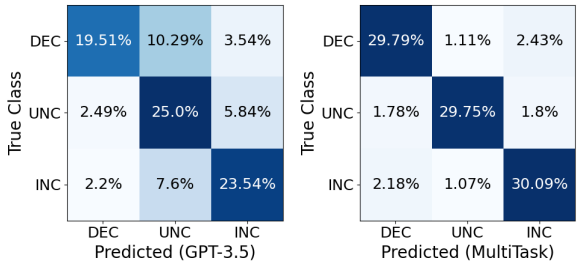

Figure 9: Confusion matrix for GPT-3.5-TURBO and MULTITASK

Pre-Fine-Tuning

To evaluate pre-fine-tuning using other temporal commonsense tasks, we compare the performance of three BERT-based models. BERT-BASE-UNCASED contains regular weights as learned during BERT's pre-training, while the two variants TACOLM (Zhou et al., 2020) and COTAK (Lynden et al., 2023) use weights trained on the two corresponding temporal commonsense datasets. We choose BERT for this evaluation as authors of both datasets publish fine-tuned BERT weights. We fine-tune the models from the published checkpoints on our own dataset in the same manner as in our main evaluation. The mean exact match score for each model is listed in Table 4. Our evaluation shows that the use of weights fine-tuned on other temporal commonsense tasks does not seem to have a positive impact on the final TVCP performance of the model. It is possible that, although the resulting embeddings of models fine-tuned on temporal commonsense tasks are more aligned with temporal properties (Zhou et al., 2020), other important information in the embeddings is lost, leading to an overall decreased performance.

Model	$\overline{\text{EM}}$
BERT-COTAK	58.2
BERT-TACOLM	59.1
BERT-BASE	**61.2**

Table 4: Results of pre-fine-tuning experiments

Error Patterns

In Figure 9, we compare the confusion matrices between true labels and GPT-3.5-TURBO and MULTITASK, respectively. We see that MULTITASK struggles more with distinguishing between the DEC and INC class, whereas GPT-3.5-TURBO classifies a rather large amount of neutral statements as non-neutral, and vice versa.

6 Conclusion and Future Work

In this work, we have introduced TVCP, a task designed to reason over the impact of context on a statement's temporal validity duration. We provide a benchmark dataset, as well as a set of baseline evaluation results. We find that the performance of fine-tuned classifiers can be improved by explicitly incorporating TVDP labels as a loss signal. Despite the impressive feats performed by foundation models, we report, similar to previous work (Bian et al., 2023; Jain et al., 2023), poor performance in the temporal commonsense domain. These findings show that users should carefully evaluate whether an LLM properly understands a given task before choosing it over smaller, fine-tuned models. We also show that models pre-fine-tuned on existing temporal commonsense tasks do not necessarily lead to better performance on TVCP.

Future work could involve using TVCP-based classifiers to collect a larger number of temporal context statements. A comparison of context-aware TVDP classifiers with previous models (Almquist and Jatowt, 2019) could emphasize the importance of accurate semantic segmentation between target- and context statements. Similarly, the use of our dataset for generative approaches could be explored, for example, in the context of generative adversarial networks. For our multitasking implementation, directions for future work could be changes to hyperparameters such as the weight of the auxiliary loss, changes to the definition of the auxiliary task (e.g., log-scaled regression or ordinal classification), or even entirely new auxiliary tasks. In the realm of LLMs, further experiments with different few-shot prompting strategies are also feasible. Finally, research into models differentiating temporal and stationary information could enhance the development and definition of future temporal validity research.

Limitations

Although we focus on creating a reproducible training- and evaluation environment, some variables are out of our control. For example, bit-wise reproducibility is only guaranteed on the same CUDA toolkit version and when executed on a GPU with the same architecture and the same number of streaming multiprocessors. This means that an exact reproduction of the models discussed in this article may not be possible. Nevertheless, we expect trends to remain the same across GPU architectures.

One of the major limitations of our approach is likely the dataset size. Although a relatively small dataset size is common in temporal commonsense reasoning, we find that our model performance still increases with the amount of training data used. The existing synthesized context statements in our dataset could be used to bootstrap an approach for automatically extracting additional samples from social media to alleviate this issue.

The data we collect is not personal in nature. However, the possibility of latent demographic biases in our data exists, for example, with respect to certain language structures or expressions used in the creation of follow-up statements. This could lead to the propagation of any such bias when the dataset is used to bootstrap further data collection, which should be considered in future work.

Our external validity is mainly threatened by two factors. First, our context statements are crowdsourced. While we apply several steps to ensure the produced context is sensible, it is unclear whether context on certain platforms, such as on social media, manifests in similar structures as in our dataset, with respect to traits such as sentence length, grammaticality, and phrasing.

Second, similar to how pre-training weights from other temporal commonsense tasks do not seem to improve the classifier performance on our dataset, the weights generated as part of our training process are likely very task-specific, and may not generalize well to other tasks or text sources.

Overall, we recommend the use of the TVCP dataset and classifiers for bootstrapping further research into combining the duration- and inference-based temporal validity tasks, as well as research into directly predicting updated temporal validity durations and improving the generalizability to different text sources, rather than for a direct downstream task application.

References

Shun Abe, Masumi Shirakawa, Tatsuya Nakamura, Takahiro Hara, Kazushi Ikeda, and Keiichiro Hoashi. 2018. Predicting the occurrence of life events from user's tweet history. In *2018 IEEE 12th International Conference on Semantic Computing (ICSC)*, pages 219–226. IEEE.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Axel Almquist and Adam Jatowt. 2019. Towards content expiry date determination: Predicting validity periods of sentences. In *European Conference on Information Retrieval*, pages 86–101. Springer.

Sitaram Asur and Bernardo A Huberman. 2010. Predicting the future with social media. In *2010 IEEE/WIC/ACM international conference on web intelligence and intelligent agent technology*, volume 1, pages 492–499. IEEE.

Prajjwal Bhargava and Vincent Ng. 2022. Commonsense knowledge reasoning and generation with pre-trained language models: a survey. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12317–12325.

Ning Bian, Xianpei Han, Le Sun, Hongyu Lin, Yaojie Lu, and Ben He. 2023. Chatgpt is a knowledgeable but inexperienced solver: An investigation of commonsense problem in large language models. *arXiv preprint arXiv:2303.16421*.

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

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6.

Bibo Cai, Xiao Ding, Bowen Chen, Li Du, and Ting Liu. 2022. Mitigating reporting bias in semi-supervised temporal commonsense inference with probabilistic soft logic. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10454–10462.

Shangkun Deng, Takashi Mitsubuchi, Kei Shioda, Tatsuro Shimada, and Akito Sakurai. 2011. Combining technical analysis with sentiment analysis for stock price prediction. In *2011 IEEE ninth international conference on dependable, autonomic and secure computing*, pages 800–807. IEEE.

Chadi Helwe, Chloé Clavel, and Fabian M Suchanek. 2021. Reasoning with transformer-based models: Deep learning, but shallow reasoning. In *3rd Conference on Automated Knowledge Base Construction*.

Taishi Hosokawa, Adam Jatowt, Masatoshi Yoshikawa, and Kazunari Sugiyama. 2023. Temporal natural language inference: Evidence-based evaluation of temporal text validity. *Proceedings of the 45th European Conference on Information Retrieval (ECIR 2023), Springer LNCS*.

Marc W Howard. 2018. Memory as perception of the past: compressed time inmind and brain. *Trends in cognitive sciences*, 22(2):124–136.

Raghav Jain, Daivik Sojitra, Arkadeep Acharya, Sriparna Saha, Adam Jatowt, and Sandipan Dandapat. 2023. Do language models have a common sense regarding time? revisiting temporal commonsense reasoning in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6750–6774.

Adam Jatowt and Ching-man Au Yeung. 2011. Extracting collective expectations about the future from large text collections. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1259–1264.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Mayuko Kimura, Lis Kanashiro Pereira, and Ichiro Kobayashi. 2021. Towards a language model for temporal commonsense reasoning. In *Proceedings of the Student Research Workshop Associated with RANLP 2021*, pages 78–84.

Yashasvi Koul, Kanishk Mamgain, and Ankit Gupta. 2022. Lifetime of tweets: a statistical analysis. *Social Network Analysis and Mining*, 12(1):101.

Niels Buus Lassen, Rene Madsen, and Ravi Vatrapu. 2014. Predicting iphone sales from iphone tweets. In *2014 IEEE 18th International Enterprise Distributed Object Computing Conference*, pages 81–90. IEEE.

Pengfei Li, Hua Lu, Nattiya Kanhabua, Sha Zhao, and Gang Pan. 2018. Location inference for non-geotagged tweets in user timelines. *IEEE Transactions on Knowledge and Data Engineering*, 31(6):1150–1165.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yafeng Lu, Robert Krüger, Dennis Thom, Feng Wang, Steffen Koch, Thomas Ertl, and Ross Maciejewski. 2014. Integrating predictive analytics and social media. In *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 193–202. IEEE.

Steven Lynden, Mehari Heilemariam, Kyoung-Sook Kim, Adam Jatowt, Akiyoshi Matono, Hai-Tao Yu, Xin Liu, and Yijun Duan. 2023. Commonsense temporal action knowledge (cotak) dataset. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM 2023)*.

James Manyika. 2023. An overview of bard: an early experiment with generative ai.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849.

Abhilash Nandy, Sushovan Haldar, Subhashis Banerjee, and Sushmita Mitra. 2020. A survey on applications of siamese neural networks in computer vision. In *2020 International Conference for Emerging Technology (INCET)*, pages 1–5. IEEE.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. Torque: A reading comprehension dataset of temporal ordering questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1158–1172.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Lis Pereira, Fei Cheng, Masayuki Asahara, and Ichiro Kobayashi. 2021. Alice++: Adversarial training for robust and effective temporal reasoning. In *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, pages 373–382.

Lis Pereira, Xiaodong Liu, Fei Cheng, Masayuki Asahara, and Ichiro Kobayashi. 2020. Adversarial training for commonsense inference. In *Proceedings of the 5th Workshop on Representation Learning for NLP (RepL4NLP-2020)*, pages 55–60. 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.

Guy Rosin and Kira Radinsky. 2022. Temporal attention for language models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1498–1508.

Guy D Rosin, Ido Guy, and Kira Radinsky. 2022. Time masking for temporal language models. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 833–841.

Shubhra Kanti Karmaker Santu and Dongji Feng. 2023. Teler: A general taxonomy of llm prompts for benchmarking complex tasks. *arXiv preprint arXiv:2305.11430*.

Lingfeng Shen, Zhuoming Liu, and Xiongtao Zhou. 2020. Forecasting people's action via social media data. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 5254–5259. IEEE.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Shane Storks, Qiaozi Gao, and Joyce Y Chai. 2019. Commonsense reasoning for natural language understanding: A survey of benchmarks, resources, and approaches. *arXiv preprint arXiv:1904.01172*, pages 1–60.

Zijun Sun, Chun Fan, Qinghong Han, Xiaofei Sun, Yuxian Meng, Fei Wu, and Jiwei Li. 2020. Self-explaining structures improve nlp models. *arXiv preprint arXiv:2012.01786*.

Hikaru Takemura and Keishi Tajima. 2012. Tweet classification based on their lifetime duration. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 2367–2370.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. *arXiv preprint arXiv:2306.08952*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Lav R Varshney and John Z Sun. 2013. Why do we perceive logarithmically? *Significance*, 10(1):28–31.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jiexin Wang, Adam Jatowt, Masatoshi Yoshikawa, and Yi Cai. 2023. Bitimebert: Extending pre-trained language representations with bi-temporal information. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 812–821.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Georg Wenzel and Adam Jatowt. 2023. An overview of temporal commonsense reasoning and acquisition. *arXiv preprint arXiv:2308.00002*.

Zonglin Yang, Xinya Du, Alexander M 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.

Changlong Yu, Hongming Zhang, Yangqiu Song, and Wilfred Ng. 2022. Cocolm: Complex commonsense enhanced language model with discourse relations. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1175–1187.

Li Zhang, Qing Lyu, and Chris Callison-Burch. 2020. Reasoning about goals, steps, and temporal ordering with wikihow. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4630–4639.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*, pages 12697–12706. PMLR.

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)*. 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.

Ben Zhou, Kyle Richardson, Qiang Ning, Tushar Khot, Ashish Sabharwal, and Dan Roth. 2021. Temporal reasoning on implicit events from distant supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1361–1371.

Bo Zhou, Yubo Chen, Kang Liu, Jun Zhao, Jiexin Xu, Xiaojian Jiang, and Qiuxia Li. 2022. Generating temporally-ordered event sequences via event optimal transport. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1875–1884.

A Twitter Collection Pipeline

To collect candidate tweets, we query the Twitter full-archive search API with the following parameters:

```
-place_country:KP lang:en
-is:retweet -is:reply
-is:quote -has:hashtags
-has:cashtags -has:links
-has:mentions -has:media
-has:images -has:video_link
```

Essentially, our goal is to collect English candidate tweets that are self-contained. This means we discard tweets that refer to other tweets (replies, mentions, retweets, or quote retweets), tweets that contain media that might provide external context (such as videos, images, links, or other types of media), and tweets that contain Twitter-specific features (hashtags, cashtags). Since Twitter's API does not allow queries based only on these conditions, we add a constraint stating that the source country of the tweet may not be North Korea, to minimize the impact on the generalizability of our target statements.

From these collected tweets, we drop duplicates and then perform basic preprocessing, including the removal of emojis, non-ASCII characters, and excess whitespace characters. As a sanity check, we remove any remaining tweets that contain http, @, or #. Most such tweets will already have been filtered by the API query, so not many tweets are lost in this step. We then apply a set of filtering steps, which are summarized in Figure 10, and detailed in the remainder of this section.

A.1 Syntactic Filtering

We first filter tweets by length, removing tweets with less than 25 or more than 200 characters. We also remove tweets containing question marks, after noting in our initial inspection that questions often have ambiguous temporal validity durations that depend on the dialogue (e.g., a question might no longer be considered temporally relevant after it has been answered). We leave such special cases to future work. We remove tweets starting with a period (.), which often manifests as ... , as well as tweets starting with a comma (,), as they indicate that a tweet may be a continuation of a prior context. Similarly, tweets ending with a colon (:) indicate that there is a dependence on some type of following context that we have not captured.

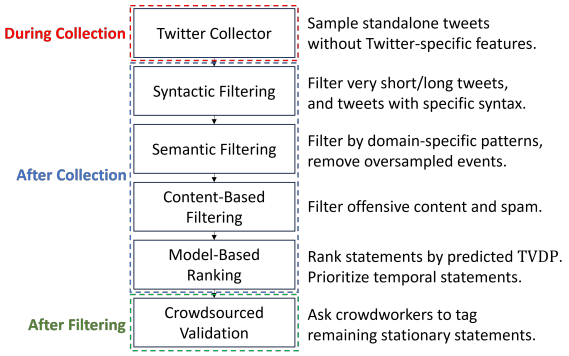

Figure 10: A summary of our tweet collection pipeline

A.2 Semantic Filtering

We devise a set of regex rules to capture specific patterns, such as recurring word repetitions, which were often associated with a small set of nonsensical spam in our sample dataset, such as "CAN WE GO BACK CAN WE GO BACK CAN WE GO BACK". Another regex string is responsible for detecting tweets containing phrases such as "Good morning" or "Have a nice Monday". These kinds of expressions are commonly found on Twitter, and, while they technically fit the task description, it is unlikely that the temporal validity duration of such a statement would be changed. Thus, we remove them from our candidate statements to increase the diversity and authenticity of generated follow-up statements. Other regex strings aim to capture common representations of threads or list iterations, such as "1/3)" or "[2/x]". Additionally, we remove tweets with more than 25% quoted text from the dataset. We find that such tweets often either consist of full quotes (often associated with figures of speech), or statements where the temporal validity of the quote is at odds with that of the rest of the statement.

A.3 Content-Based Filtering

To filter offensive content, we use a binary classification model based on RoBERTa, which can be downloaded and run via the Huggingface transformers library under the name "cardiffnlp/twitter-roberta-base-offensive". We also use the word-list-based *better-profanity* Python library[3]. To min-

[3]https://pypi.org/project/better-profanity/

imize offensive content in our dataset, we only keep tweets that do not contain any profanity from the word list and that are considered non-offensive by the transformer model. To filter out additional spam, we use a BERT-TINY-based binary classification model trained on SMS spam, which can be found under the name "mrm8488/bert-tiny-finetuned-sms-spam-detection" in the Huggingface transformers library. We find that most items have a relatively low probability of being spam under this model, with the upper fence being roughly 0.07. Thus, we chose 0.07 as a cut-off point, removing all samples that exceed this probability. While not very sophisticated, this model works well for detecting texts containing hashes, Bitcoin addresses, or other types of data that do not fit into our task description.

A.4 Model-Based Ranking

We apply an ensemble of models based on the previously mentioned COTAK (Lynden et al., 2023) and ALMQUIST2019 (Almquist and Jatowt, 2019) datasets. The authors of the COTAK dataset publish their model for estimating action effect durations, which we retrieve from the Huggingface transformers library under the name "mrfriedpotato/effect". For the ALMQUIST2019 dataset, we train a neural network based on the pooler output of BERT-BASE-UNCASED, initialized with epoch 2 weights from TACOLM (Zhou et al., 2020). The model parameters are frozen, while a single linear layer predicting the output class is trained with a dropout probability of 0.1. ADAMW (Loshchilov and Hutter, 2018) was used as an optimizer with a learning rate of $1e-5$. The corresponding model achieved 0.74 Micro-F1 and 0.69 Macro-F1 on an 80-10-10 train-val-test split after just a few epochs, outperforming non-transformer-based methods proposed in the original paper.

We rank statements by the averaged probability that they are not classified as *longer (than days)* in the COTAK-based model, and are not classified as *months* or *years* in the ALMQUIST2019-based model. A statement that either model classifies as one of the classes mentioned above is automatically assigned a score of -1. For all other statements, the softmax scores of the remaining classes are summed up, and the average of the two summed probabilities is used as the score. We then sort the resulting dataset by this score and prioritize the highest-scoring statements for crowdsourcing.

B Crowdsourcing Definitions

In this section, we provide details on the crowdsourcing implementation. As noted, we use Amazon Mechanical Turk to collect crowdsourced data from participants.

B.1 Temporal Validity Duration Estimation

We assume the average layman is not familiar with the term *temporal validity*. Thus, we define the task as "determining how long the information within the tweet remains relevant after its publication", i.e., for how long the user would consider the tweet timely and relevant. We provide the option *no time-sensitive information* to tag any stationary statements that were not removed during preprocessing. The task is otherwise a relatively straightforward classification task. We split our dataset into batches of 10 samples that are grouped into a single *human intelligence task* (HIT). For each HIT, we offer a compensation of USD0.25, based on an estimated 6-9 seconds of processing time per individual statement (i.e., 60-90 seconds per HIT). Figures 11 to 14 show the crowdsourcing layout.

B.2 Follow-Up Content Generation

Compared to the temporal validity duration estimation task, the follow-up content generation task requires a much more robust understanding of the overall concept of temporal validity and the respective semantic roles of the target- and follow-up statements. Hence, we focus on providing a more detailed explanation of the task. Figures 15 to 17 show the crowdsourcing setup. The detailed instructions tab is not listed due to its length, but contains instructions that can also be found in the code repository as part of the qualification test. Notably, we labelled the target statement as *context tweet* in this crowdsourcing task to emphasize that participants should not alter this statement directly, as this was a problem that occurred somewhat frequently during pilot tests. This contrasts with our formal definition of TVCP, where providing context is the role of the follow-up statement.

Each HIT requires participants to provide three follow-up statements, one for each TVCP class (DEC, UNC, INC), as well as the corresponding updated TVDP labels. For each HIT, we offer a compensation of USD0.35. We base our compensation on an estimated 30–40 seconds of processing time per follow-up statement (i.e., 90–120 seconds per HIT) due to the creative writing involved.

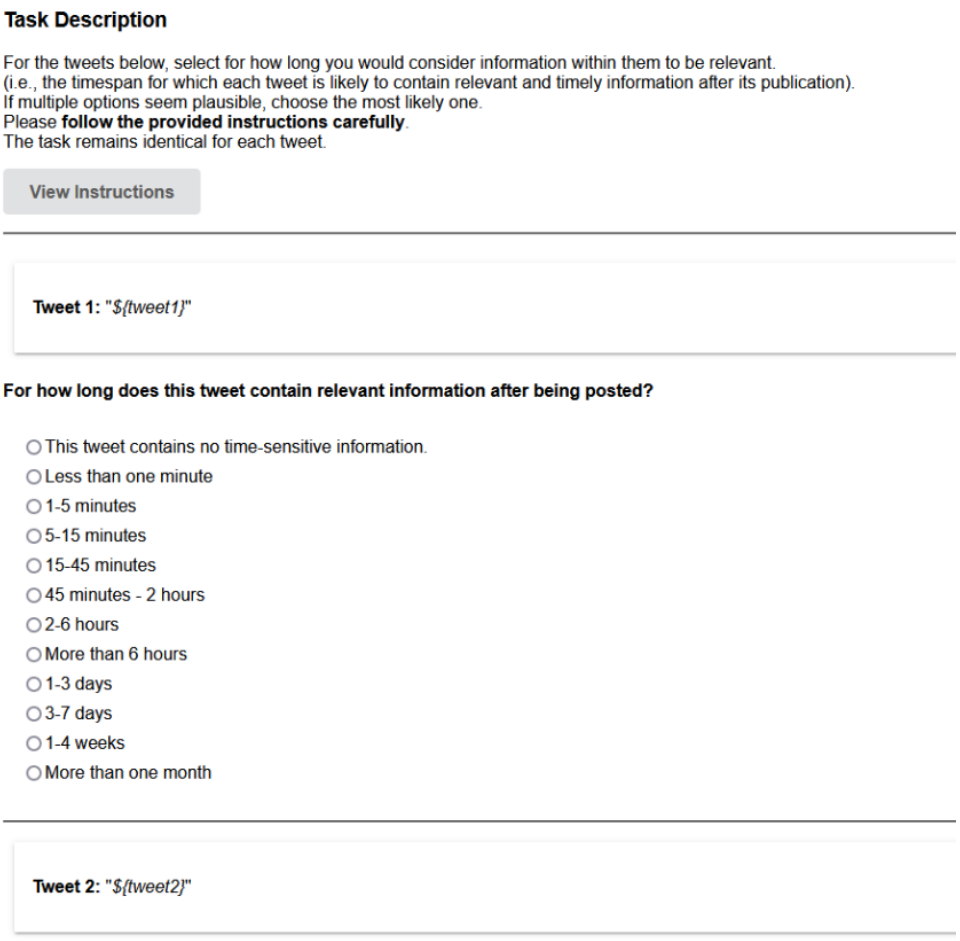

Figure 11: The interface of the temporal validity duration estimation task

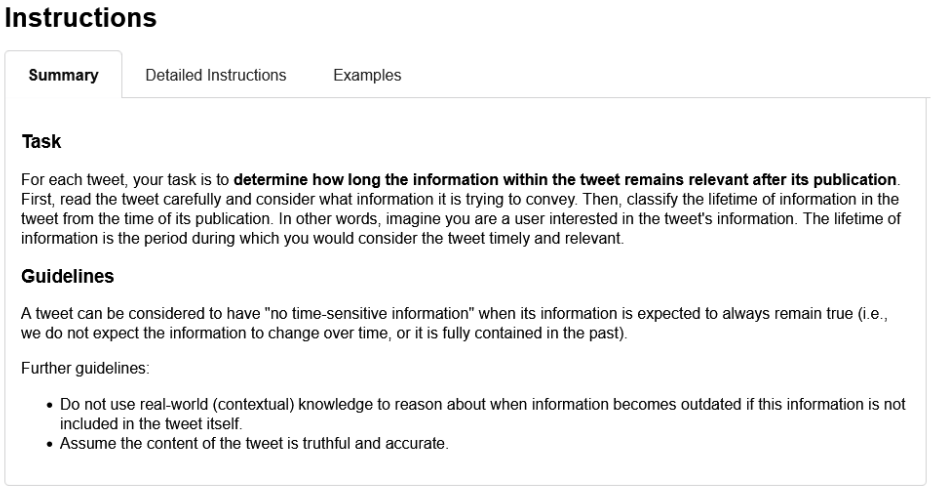

Figure 12: The summary section of the temporal validity duration estimation task guidelines

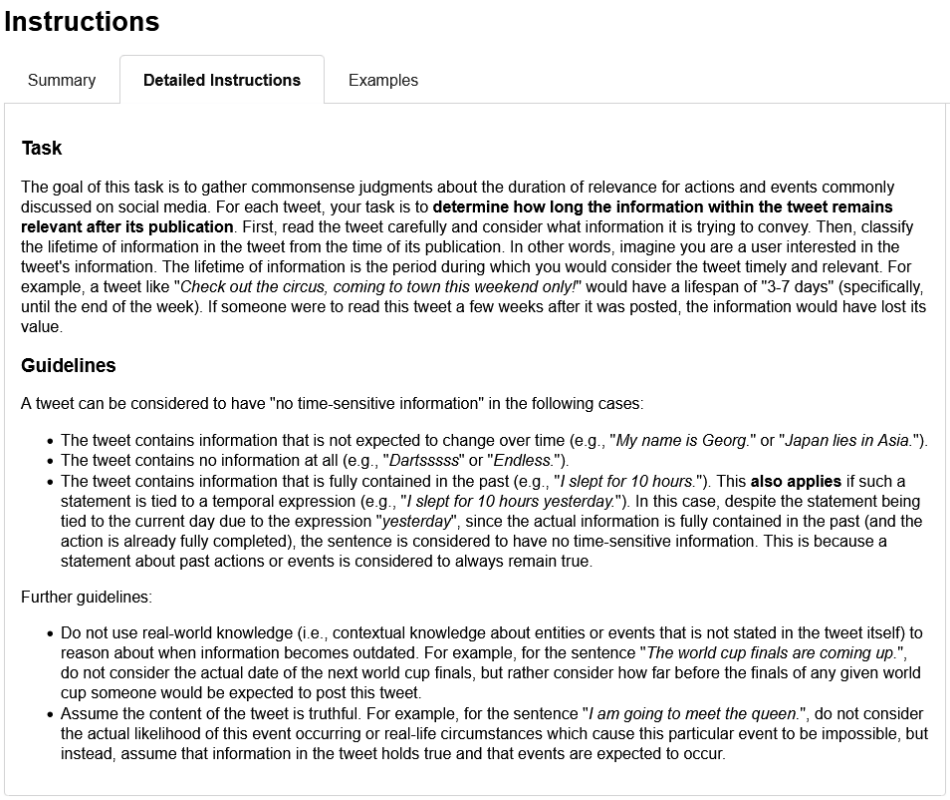

Figure 13: The detailed description of the temporal validity duration estimation task guidelines

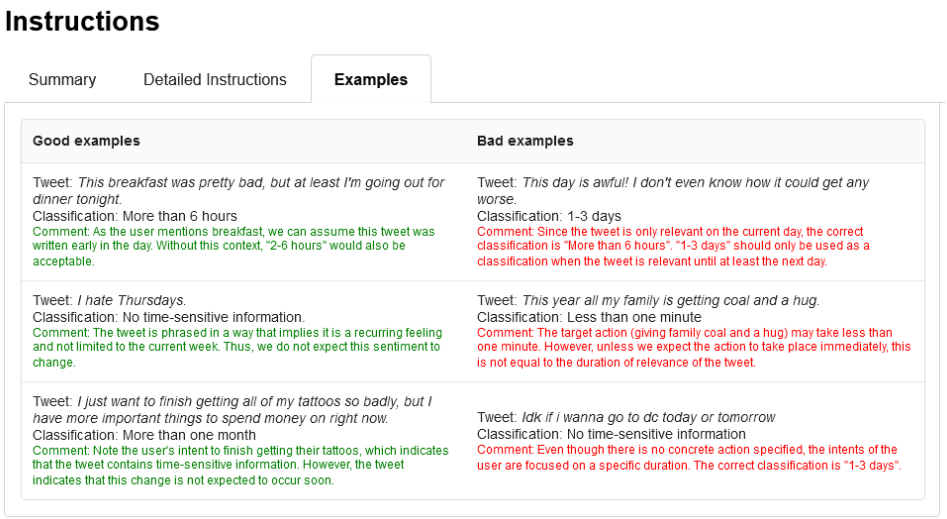

Figure 14: The examples section of the temporal validity duration estimation task guidelines

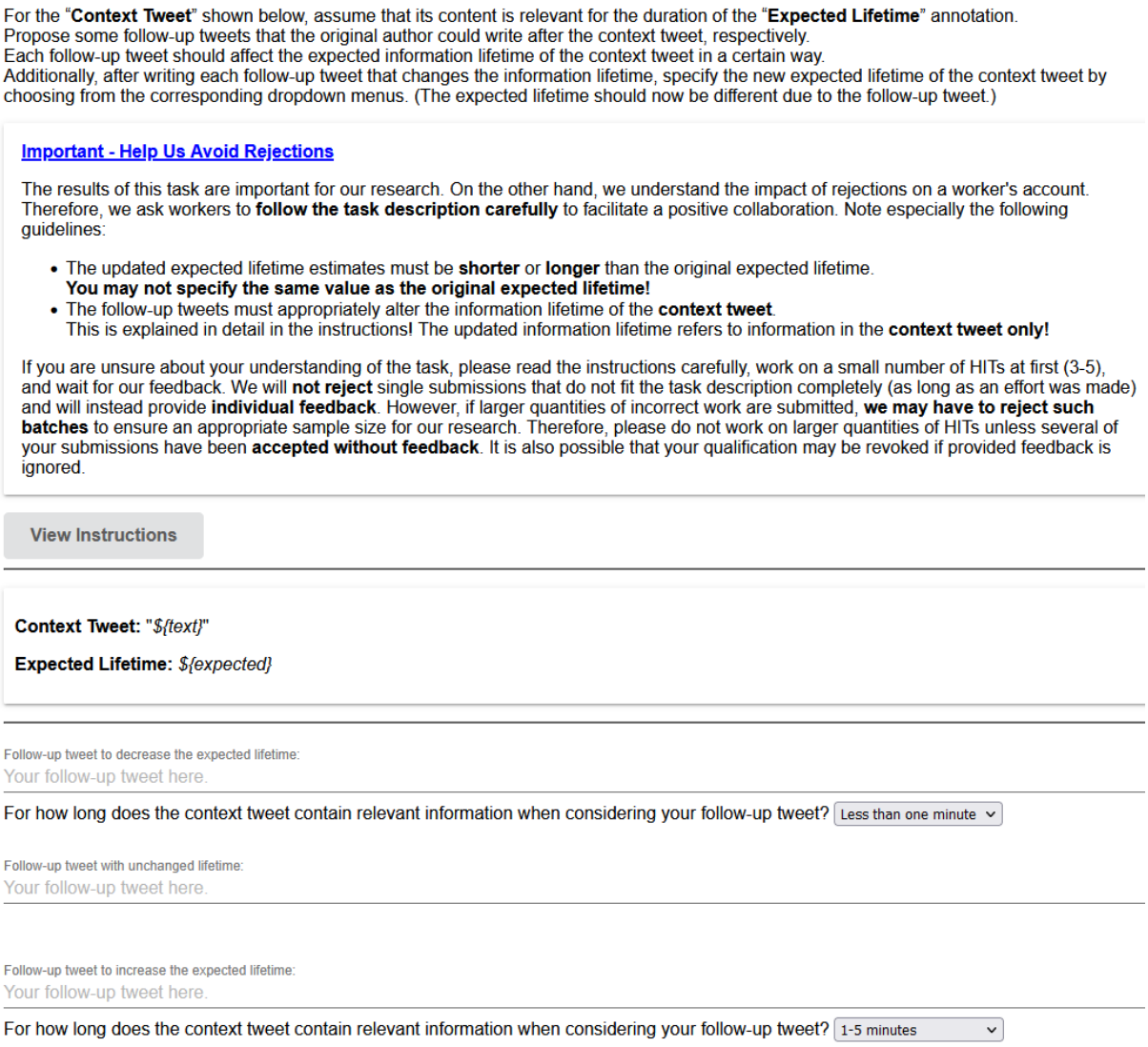

Figure 15: The interface of the follow-up content generation task

Instructions

Summary | Detailed Instructions | Examples

Task

In this crowdsourcing task, you are given a context tweet with an "expected lifetime" that indicates how long the information in the tweet will be relevant. Your task is to write three follow-up tweets:

- One where the expected lifetime of information in the context tweet **decreases**.
- One where the expected lifetime of information in the context tweet **remains unchanged**.
- One where the expected lifetime of information in the context tweet **increases**.

For the follow-up tweets that change the expected lifetime, you must also provide an updated lifetime estimate for the context tweet. Note that this new estimate is a period starting at the creation of the context tweet, **not** the follow-up tweet. Additionally, it must be a different class than the original expected lifetime (at least the adjacent shorter/longer class).

Guidelines

- Do not change the context tweet itself. Write follow-up tweets instead.
- You may not specify the same value as the original expected lifetime for your updated lifetime estimates.
- Focus on changing the lifetime of information in the context tweet.
- Give your best effort when the context tweet is unclear.
- Try to avoid using explicit temporal expressions.
- Be creative and come up with varied scenarios that change the expected information lifetime.

Possible Reasons for Rejection

We appreciate your contributions to our crowdsourcing tasks and strive to avoid rejecting work. However, in cases where the work submitted does not meet the requirements of the provided task, we may be unable to issue payment. **Work may be rejected if you submit a large number of HITs that do not follow the task description**. Most notably, some reasons for rejection may be:

- The work submitted does not adhere to the task description, especially the guidelines highlighted within the HIT interface and the instruction summary.
- The work appears to be "low-effort" (e.g., simply stating that an action will take a certain amount of time without providing further context).
- The work is written in poor English. While perfect grammar is not required, the level of English should at least match that of the context tweet.
- The provided updated lifetime estimates do not follow the task description. For instance, if the objective is to increase the lifetime of information, the work may be rejected if the updated lifetime estimate is not longer than the original estimate.

Figure 16: The summary section of the follow-up content generation task guidelines

Instructions

Summary | Detailed Instructions | **Examples**

Good examples	Bad examples
Context Tweet: *"Going to the gym after work today!"* **Expected Lifetime:** More than 6 hours	**Context Tweet:** *"Going to the gym after work today!"* **Expected Lifetime:** More than 6 hours
Follow-up to decrease the expected lifetime: *"Actually, I'll get a quick workout in during my lunch break at the gym next door."* New expected lifetime: *2-6 hours* **Why?** The main information in the context tweet (going to the gym / working out) remains valid, but the action will now occur within a more immediate timeframe.	Follow-up to decrease the expected lifetime: *"I'm so sore from yesterday's workout that I can barely move. Skipping the gym today."* New expected lifetime: *Less than one minute* **Why?** Consider the difference between an action that does not occur at all, and an action that occurs very quickly. In this example, the context tweet's information does not apply.
Follow-up with unchanged lifetime: *"I think I'll try out a new HIIT workout."* **Why?** The follow-up tweet relates to the context tweet, but does not change the expected lifetime of information.	Follow-up with unchanged lifetime: *"I think I'll get pizza for dinner tonight."* **Why?** In the unchanged lifetime task, there should still be some topical connection between the follow-up and the context tweet.
Follow-up to increase the expected lifetime: *"Have to work overtime today. The gym will have to wait until tomorrow."* New expected lifetime: *1-3 days* **Why?** As the author confirms that the planned action will still take place, we consider the information lifetime in the context tweet as continuously valid until this new date.	Follow-up to increase the expected lifetime: *"The gym is closed today due to a maintenance issue. Guess I'm not going."* New expected lifetime: *1-3 days* **Why?** A follow-up tweet cancelling plans can only be considered an appropriate follow-up when it is clear the action will still take place at a later date, which is not the case in this example.

Figure 17: The examples section of the follow-up content generation task guidelines

B.3 Discouraging Dishonest Activity

In initial pilot runs, we find that many submissions are the result of spam, dishonest activity, or a complete lack of task understanding, with many provided annotations being inexplicable by common sense. To increase the quality of work on both tasks, we introduced three measures.

First, we required participants to have an overall approval rate of 90% on the platform, as well as 1,000 approved HITs. Without these requirements, the amount of blatant spam (e.g., copy-pasted content) increases significantly.

Second, we devised qualification tests for both tasks. Participants had to determine the temporal validity durations for a set of sample statements to work on the temporal validity duration estimation task, and determine the correctness of follow-up statements and their updated duration labels to work on the follow-up content generation task.

Finally, we vet all participants' responses individually up to a certain threshold. For each task, we manually verify the first 20 submissions of each annotator on their quality. We provide feedback and manually adapt submissions when they are partially incorrect. If submission quality is appropriate by the time a participant reaches 20 submitted HITs, we consider them as trusted, and only spot-check every 5th submission thereafter. If submission quality does not sufficiently improve at this point, we prohibit the participant from further working on the task.

C Evaluation Setup

C.1 Fine-Tuning Strategy

We perform several experiments to improve the model setup for our fine-tuned baselines, BERT and RoBERTa. First, for each model, we evaluate two separate pipelines. The TransformerClassifier pipeline concatenates both statements of a sample before embedding them jointly, whereas the SiameseClassifier (Bromley et al., 1993; Nandy et al., 2020) pipeline generates a separate embedding for the target- and context statement, and combines them to form a hidden representation $[h_{s_t}, h_{s_f}, h_{s_t} - h_{s_f}, h_{s_t} \otimes h_{s_f}]$.

We perform hyperparameter testing regarding dropout probability before the classification layer (0.1, 0.25, 0.5), the base learning rate (1e-2, 1e-3, 1e-4), and whether to freeze embedding layers (i.e., training only intermediary and classification layers). For both BERT and RoBERTa in the frozen and unfrozen setting, we perform grid-search over the learning rate and dropout probability.

For both hyperparameter optimization and model training, we use the AdamW optimizer (Loshchilov and Hutter, 2018) with ε = 1e-8, $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight_decay = 0.01 and optimize for cross-entropy loss. For hyperparameter testing, we use a predefined train-val-test split (80%/10%/10%) rather than the five-fold evaluation in Section 5. In both cases, if the validation EM score does not exceed the best previously observed value for 5 consecutive epochs, we stop training early. The model epoch with the best validation EM score is used for evaluating the test set.

Table 5 shows the three best-performing configurations for BERT and RoBERTa in the freeze and nofreeze settings, respectively, on the TransformerClassifier pipeline. Table 6 shows the same results for the SiameseClassifier pipeline.

Model	DO	LR	#Epochs	EM
BERT-nofreeze	0.25	1e-4	5	0.613
BERT-nofreeze	0.10	1e-4	6	0.548
BERT-nofreeze	0.50	1e-4	4	0.548
BERT	0.25	1e-4	17	0.321
BERT	0.10	1e-4	8	0.315
BERT	0.10	1e-3	10	0.304
RoBERTa	0.25	1e-3	14	0.262
RoBERTa	0.10	1e-4	16	0.256
RoBERTa	0.50	1e-3	15	0.238
RoBERTa-nofreeze	0.25	1e-3	1	0.000
RoBERTa-nofreeze	0.50	1e-3	1	0.000
RoBERTa-nofreeze	0.10	1e-4	1	0.000

Table 5: Best three models for each of the proposed configurations in the TransformerClassifier pipeline

Model	DO	LR	#Epoch	EM
BERT-nofreeze	0.25	1e-4	7	0.589
BERT-nofreeze	0.10	1e-4	4	0.577
BERT-nofreeze	0.50	1e-4	2	0.565
RoBERTa	0.10	1e-4	21	0.548
RoBERTa	0.50	1e-4	13	0.518
RoBERTa	0.25	1e-4	17	0.512
BERT	0.50	1e-4	9	0.387
BERT	0.25	1e-3	8	0.357
BERT	0.25	1e-4	5	0.339
RoBERTa-nofreeze	0.25	1e-3	1	0.000
RoBERTa-nofreeze	0.50	1e-3	1	0.000
RoBERTa-nofreeze	0.10	1e-4	1	0.000

Table 6: Best three models for each of the proposed configurations in the SiameseClassifier pipeline

The most notable finding appears to be that RoBERTa gets stuck in a false minimum of predicting a constant class when embedding layers are unfrozen, leading to an accuracy of 0.33 and an

EM of 0. Hence, we freeze embedding layers for RoBERTa in our main evaluation. RoBERTa-based models with frozen embedding layers tend to have a worse baseline performance, but have a higher relative improvement when switching to the SiameseClassifier implementation. We hypothesize that RoBERTa's sentence embedding token, <s>, may contain less information about the full sequence than BERT's [SEP] token, due to the lack of a next-sentence-prediction task during pre-training.

For SelfExplain, we use the originally proposed learning rate of $2e-5$ and no dropout. The evaluation setup is otherwise identical. The final layer of all models before classification has a dimensionality of 768. All models were trained and evaluated on an MSI GeForce RTX 3080 GAMING X TRIO 10G GPU using CUDA 11.7. Training and evaluation of the models, as well as hyperparameter tests, took around 15 GPU hours.

C.2 Few-Shot Prompting Strategy

For models evaluated via few-shot prompting, we first provide the following system prompt:

> "You are a language model specialized in reasoning over temporal common sense. You know that the temporal validity duration of a statement is the duration for which said statement contains relevant and current information after its creation. Information that takes place in the future, such as "I will take a shower at 8 p.m.", is considered valid from the point of statement creation until the information has fully resolved.
>
> Your task is to determine the impact of a context statement on the temporal validity duration of a target statement. The user will provide both statements. When a statement can be interpreted in multiple ways, assume the most likely interpretation is the correct one.
>
> To solve the task effectively, follow the steps outlined below:
>
> 1. Ignoring the context statement, determine the temporal validity duration of the target statement. Your estimate must match one of the following labels: [less than one minute, 1-5 minutes, 5-15 minutes, 15-45 minutes, 45 minutes - 2 hours, 2-6 hours, more than 6 hours, 1-3 days, 3-7 days, 1-4 weeks, more than one month]. Select exactly one class and explain why it is the most fitting.
>
> 2. Once again, determine the most likely of the above labels to match the temporal validity duration of the target statement, but this time, include any information from the context statement that may influence the class label. Similar to step 1, explain why you chose the class.
>
> 3. Compare the two class labels generated in step 1 and step 2 to infer the change in the temporal validity duration. Select the appropriate class label [decreased, unchanged, increased], depending on if the introduction of the context statement decreased, remained neutral to, or increased the temporal validity duration of the target statement. You must surround this class label with backticks (`)."

Instead of prompting the model to classify the sample directly, we ask it to provide an explanation for its decision based on chain-of-thought reasoning. Wei et al. (2022) show that chain-of-thought prompting significantly increases several types of reasoning capabilities, including commonsense, in LLMs.

After the system prompt, we provide nine sample conversation turns, each consisting of a user prompt and a corresponding sample response by the system. For each TVCP class, we provide three samples. Each sample classification begins with the following user prompt.

> "Please determine the impact of the context statement on the temporal validity duration of the target statement.
>
> Target Statement: [s_t]
> Context Statement: [s_f]"

Afterwards, an explanation is provided by the assistant role. All few-shot samples are listed below, sorted by their target class. In the actual prompt, we apply a fixed perturbation to the order of few-shot samples to avoid recency bias in the model prompting (e.g., if all INC class samples were placed at the end of the few-shot prompt, this might encourage the model to assign the INC class more frequently,

Zhao et al., 2021). The temperature of the models is set to 0 for all generative experiments.

Target Statement: "My sister: We should go out for a drink. Just ONE. Us: *end up getting drunk*"
Context Statement: "Just found out I have an early morning meeting tomorrow"
Explanation: "The target statement suggests a night out, which has already started. Such an occasion may still continue for several hours after the statement was made. For this reason, the most fitting label is 2-6 hours. The context statement notes an early morning meeting on the following day. Upon finding out about this meeting, it is likely that the author would cut their night out short. Although it is not clear how much time has passed between the two statements, a reasonable assumption is that the author would discover such a meeting and subsequently cease the activity within the next 15-45 minutes. For this reason, the most fitting label is 15-45 minutes. As the information in the context statement decreases the temporal validity duration of the target statement from 2-6 hours to 15-45 minutes, the final label is 'decreased'"

Target Statement: "Cab was early, so here I am at Midway with an hour to spare. And get Dunkin' Donuts does not actually have any donuts."
Context Statement: "I might just take the earlier train then, it's a flex ticket anyway."
Explanation: "The target statement states that the author has an hour to spare at Midway. For this reason, the most fitting label is 45 minutes - 2 hours. The context statement specifies that the author could be taking an earlier train, reducing the amount of time they have to spare at Midway. As they do not seem to be in a hurry, the most fitting updated label is 15-45 minutes. As the information in the context statement decreases the temporal validity duration of the target statement from 45 minutes - 2 hours to 15-45 minutes, the final label is 'decreased'"

Target Statement: "I have 1 extra LA Bakeshop Cheesebread /Spanish Bread Mix. DM if you want it."
Context Statement: "Whoops, sorry. . . gone in seconds."
Explanation: "The target statement contains an offer regarding some free bread mix. Depending on the reach of the author, it is likely that someone would be interested in this offer relatively quickly. For this reason, the most fitting label is 15-45 minutes. The context statement states that the offer was taken within seconds. If this statement can be taken at face value, it suggests the most fitting label is less than one minute. As the information in the context statement decreases the temporal validity duration of the target statement from 15-45 minutes to less than one minute, the final label is 'decreased'"

Target Statement: "taxi going home. back to reality."
Context Statement: "How can a week long vacation feel like a single day. . . ?"
Explanation: "The target statement suggests that the author is on a taxi ride home. The typical duration of a taxi ride may be between 15-45 minutes. For this reason, the most fitting label is 15-45 minutes. The context statement provides additional information about where the author is coming from, and about the duration of their vacation. However, the duration of the taxi ride is unaffected by this information. For this reason, the most fitting label is still 15-45 minutes. As the information in the context statement does not change the temporal validity duration of the target statement, the final label is 'unchanged'"

Target Statement: "I swear I woke up with the mindset to have a great day"
Context Statement: "Meditation really seems to be working. . . maybe I should stick with it"
Explanation: "The target statement refers to a mindset regarding the current day. As a result, the most fitting label is More than 6 hours. The context statement reveals meditation as a possible cause for the author's current mindset. The author contemplates sticking with meditation to improve their mood over a longer duration. However, the context statement is not clear on if the author will follow through on this thought, and the process of meditation is not necessarily linked to the author's long-term feelings. For this reason, the most fitting label is still More than 6 hours. As the information in the context statement does not change the temporal validity duration of the target statement, the final label is 'unchanged'"

Target Statement: "BTW I'm playing splatoon almost all day, come poke me"

Context Statement: "I can play this game for hours. Maybe I should go pro"
Explanation: "According to the target statement, the author plans on playing the game Splatoon almost all day. For this reason, the most fitting label is More than 6 hours. In the context statement, the author further professes their enjoyment for the game and contemplates possible future plans. However, this information is not likely to impact the duration for which the playsession in question will continue. For this reason, the most fitting label is still More than 6 hours. As the information in the context statement does not change the temporal validity duration of the target statement, the final label is 'unchanged'"

Target Statement: "I'm watching Perfect Match and I feel like this chic Francesca is so selflish"
Context Statement: "Why did they make her the focus of the double runtime special???"
Explanation: "Perfect Match is a TV series with a typical episode runtime between 38-65 minutes. According to the target statement, the author has already started watching the episode, so some of the episode's runtime has likely already passed. For this reason, the most fitting label is 15-45 minutes. The context statement notes that the episode is a double runtime special. In this case, it is likely that the remaining runtime exceeds 45 minutes. For this reason, the most fitting label is 45 minutes - 2 hours. As the information in the context statement increases the temporal validity duration of the target statement from 15-45 minutes to 45 minutes - 2 hours, the final label is 'increased'"

Target Statement: "I'm about to read some tech news really quick then enjoy dinner."
Context Statement: "Oh, I forgot we had a reservation at that fancy dinner place later today. . . I guess I'll get by with chips for now"
Explanation: "Reading tech news and subsequently enjoying dinner, as suggested by the target statement, is likely to take longer than 45 minutes, but unlikely to take longer than 2 hours. For this reason, the most fitting label is 45 minutes - 2 hours. The context reveals that the author has a dinner reservation later in the day, and will only be eating chips for the time being. Since the dinner is already mentioned in the target statement, it remains valid until this event takes place. As the reservation takes place later in the day, the most fitting label is 2-6 hours. As the information in the context statement increases the temporal validity duration of the target statement from 45 minutes - 2 hours to 2-6 hours, the final label is 'increased'"

Target Statement: "just patiently waiting for my bff to get here"
Context Statement: "He'll be flying in from the states next Monday. So excited!!!"
Explanation: "The target statement states that the author is waiting on the arrival of their bff. With no further information, it is reasonable this refers to an imminent situation. For this reason, the most fitting label is 15-45 minutes. The context statement reveals that the author's BFF is in fact flying in from abroad in the following week, thus, the event is not imminent. Since the context statement mentions this event takes place on the next Monday, it is not possible for more than 7 full days to pass before this event occurs. As a result, the most fitting label is 3-7 days. As the information in the context statement increases the temporal validity duration of the target statement from 15-45 minutes to 3-7 days, the final label is 'increased'"
