Probing Language Models for Understanding of Temporal Expressions

Shivin Thukral and **Kunal Kukreja** and **Christian Kavouras**
Department of Linguistics
University of Washington
{shivin7, kkukreja, cdkavour}@uw.edu

Abstract

We present three Natural Language Inference (NLI) challenge sets that can evaluate NLI models on their understanding of temporal expressions. More specifically, we probe these models for three temporal properties: (a) the order between points in time, (b) the duration between two points in time, (c) the relation between the magnitude of times specified in different units. We find that although large language models fine-tuned on MNLI have some basic perception of the order between points in time, at large, these models do not have a thorough understanding of the relation between temporal expressions.

1 Introduction

While contextualized embeddings obtained from recent transformer-based models such as BERT (Devlin et al., 2019) have proven to contain a lot of semantic and syntactic information about the tokens they encode, recent studies have shown that there are still gaps in their understanding (Rogers et al., 2020). On the semantic side, for instance, BERT struggles with representations of numbers (Wallace et al., 2019) and cannot reason based on its world knowledge (Rogers et al., 2020). Work in NLI has also developed challenge sets showing that the reported performance of these language models on various tasks can be exaggerated (McCoy et al., 2019), and they rely on lexical cues in the dataset instead of actual language comprehension.

Our work explores the grasp of such models on the relation between temporal expressions. Temporal expressions, or time expressions, in text are a sequence of tokens that denote time, such as a point in time (6 May 1980, Monday, 12 PM) or duration (7 minutes, 5 years, 2 months). More specifically, we try to determine whether these models capture the ordering and duration relationships between different points in time. We also analyze if these models can reason about durations specified in different units. Recognition of temporal expressions has had applications in timeline construction (Do et al., 2012; Leeuwenberg and Moens, 2018) and clinical analysis (Bethard et al., 2015) previously, and can be beneficial for dialogue assistants in scheduling reminders and meetings, which shapes our motivation behind conducting such an analysis. We evaluate these models on the above temporal properties by presenting three NLI challenge sets.

Our experiments demonstrate that language models such as RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2021) fine-tuned on existing large NLI datasets are unable to completely reason about the ordering and duration between temporal expressions. We further analyze the examples and find that while these models recognize whether a point in time lies within an interval, they cannot capture other relations between time instances and durations[1].

2 Related Work

Much work has been done on the extraction of events, temporal expressions (Chen et al., 2019; Ding et al., 2019), and the temporal relations between the two. TimeBank (Pustejovsky et al., 2003b) was one of the first annotated corpora for this task. It utilized the TimeML (Pustejovsky et al., 2003a) standard for annotation. TempEval-1 (Verhagen et al., 2007), TempEval-2 (Verhagen et al., 2010), and TempEval-3 (UzZaman et al., 2013) are shared tasks created for evaluating models on various temporal properties, and most methods used were traditional rule-based (Strötgen and Gertz, 2010; Ning et al., 2018) or grammar-based (Lee et al., 2014) solutions.

Various corpora have been developed that test for different temporal properties. Vashishtha et al. (2019) map events to their fine-grained duration, and event pairs to their relative timelines. Naik et al.

[1]Code and data available on GitHub

Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pages 396–406
Online, November 11, 2021.

(2019) create additional annotations in the existing TimeBank-Dense corpus (Cassidy et al., 2014) for discourse-level temporal ordering. Ning et al. (2020) create a reading comprehension dataset that tests for temporal ordering. Zhou et al. (2019) test for various temporal commonsense properties using a multiple-choice question-answering dataset. Vashishtha et al. (2020) recast existing temporal datasets into NLI format to test for temporal ordering and duration.

Our goal is to create similar datasets to probe for a semantic understanding of temporal expressions in pre-trained language models. We create these datasets in an NLI format and use them to evaluate NLI models trained on MNLI (Williams et al., 2018), which is a generic NLI dataset. We choose MNLI because it is large and diverse. The dataset contains time terms in 36% of the development instances, including examples containing temporal expressions like months and days of the week. We investigate whether these examples are sufficient for a general perception of temporal expressions.

To our knowledge, there has been little work investigating the implicit understanding of time expressions in pre-trained large language models. The most similar work to ours is Vashishtha et al. (2020). They produce five NLI datasets recast from existing temporal reasoning corpora and test NLI models for event duration (how long an event lasts) and event ordering (how events are temporally arranged). However, there are some key differences:

- Our focus is to investigate the temporal properties of ordering and duration for explicit *time expressions*, and not for *events* in a sentence.
- We analyze whether language models can reason about more fine-grained duration (e.g., whether an event takes exactly 5 hours) where as they analyze reasoning about more coarse-grained duration (e.g., whether an event takes place in the order of hours or days).
- We also investigate whether language models can figure out commonplace conversions among adjacent units of time.
- We introduce numerous variations in our data creation process about how the time expressions are inserted and draw conclusions from how these variations affect performance.

List Type	List Range
hour (12 hr)	*12 AM, ..., 11 PM*
hour (24 hr)	*00:00, 01:00, ..., 23:00*
weekday	*Sunday, ..., Saturday*
month-day	*1st, 2nd, ..., 28th*
month (full name)	*January, ..., December*
month (abbreviated)	*Jan, Feb, ..., Dec*
year	*1900, 1901, ..., 2000*

Table 1: Different lists of temporal expressions

3 Dataset Creation

We construct three NLI datasets that aim to test different relations between temporal expressions. The datasets use templates from a manually curated list of 71 events, labeled with their *temporal occurrence* (when the event is likely to occur) and *temporal duration* (how long the event is expected to last) values. For instance:

Template: I went to Paris
Occurrence: day, month, year
• I went to Paris on Monday.
• I went to Paris in March.
• I went to Paris in 2010.
Duration: hours, days
• I visited Paris from 10 AM to 9 PM.
• I visited Paris from Mon to Wed.

Each temporal unit corresponds to some list(s) spanning different magnitudes of time (Table 1), which are used during NLI pair creation.

3.1 Set I: Temp-Order

We create this NLI challenge set to test whether language models recognize the relationship of ordering between two distinct temporal expressions. We frame this in the NLI format by having the premise mention an event occur at a particular time instance, while the hypothesis mentions the same event but occurring at a different time instance:

Premise : They got married **in March**.
Hypothesis : They got married **before July**.
Label : Entailment

We start constructing a basic NLI pair by choosing a sentence template from the list of events. Based on the *temporal occurrence* label of the event, one of the lists from Table 1 is chosen, and two time instances are sampled from that list with

	Premise	Hypothesis	Label
a)	He left his job *at 12 PM*.	He left his job *before 5 PM*.	E
b)	*At 12 PM*, he left his job.	*Before 5 PM*, he left his job.	E
c)	He will leave his job *at 12 PM*.	He will leave his job *before 5 PM*.	E
d)	He left his job *after 12 PM*.	He left his job *after 9 AM*.	E
e)	He left his job *after 12 PM*.	He left his job *before 5 PM*.	N
f)	He left his job *after 12 PM*.	He left his job *before 9 AM*.	C
g)	He left his job *at 12 PM*.	He left his job *before 17:00*.	E
h)	He left his job *in February*.	He left his job *after Apr*.	C
i)	He left his job *in October 2011*.	He left his job *after Jan 2011*.	E
j)	He left his job *on 21st Sep 2013*.	He left his job *before 23rd Sep 2012*.	C

Table 2: Variations in NLI pairs for ordering of temporal expressions (E → entailment, C → contradiction, N → neutral). *a)* is the basic construction; *b), c)* is with the variation in event template; *d), e), f)* are when premise uses a relative preposition to allow the event to happen in a time interval; *g), h)* are examples of choosing time instances from two different lists; *i), j)* are generation of more specific dates using months and month-days with years.

replacement. For the premise, the first time instance is attached so that the event happens precisely at this time instance. For the hypothesis, we randomly choose a *relative ordering* between *'before'* and *'after'* and attach it to the template event and the second instance. Since the premise claims that the event occurs at an exact point in time while the hypothesis claims that the event happens in a specific time interval, the premise time instance either lies inside the hypothesis time interval or it does not, generating the labels of *entailment* or *contradiction* correspondingly.

During label generation, we have assumed that both time instances lie in the same cycle (e.g., two *weekdays* lie in the same week). However, for cases where the two time instances are close across consecutive cycles, the automated label generated this way might be considered conventionally wrong:

Premise: The concert starts **at 2 AM**.
Hypothesis: The concert starts **before 11 PM**.
Label: Entailment

To reduce the number of such edge cases in the dataset, we do not allow sampling of time instances that are more than half the length of the list far apart (e.g., for within a day, the distance between two *hours* will be at most 12).

We also introduce some variations in the sentence generation process to analyze the sensitivity of the models. Firstly, we tweak the event template by changing its position in the sentence (Table 2 *b*) and by switching it to future tense (Table 2 *c*). Secondly, we allow the premise event to also occur over an interval of time rather than a point in time (Table 2 *d, e, f*). To generate labels for these cases, the criteria we follow is that the pair is an *entailment* if the premise time interval is completely included in the hypothesis time interval (*temporal inclusion*), a *contradiction* if there is no overlap between the two (*temporal precedence*), and *neutral* otherwise. Moreover, we allow the premise and hypothesis to sample points in time from different lists when possible (Table 2 *g, h*). We also generate more specific dates by combining *months* and *month-days* with *years* (Table 2 *i, j*) to see if the language models are still able to reason about the difference in their ordering.

We construct separate train and test datasets, using 53 templates for the train split and 18 templates for the test split. We have 11 different ways of choosing the two time instances: seven ways of choosing both from the same list (Table 1) and four ways of choosing from different lists (Table 2 *g-j*). We choose the two time instances for each template based on its *temporal occurrence* label and run it for five iterations, which results in a train dataset of 16,980 instances and test dataset of 6,140 instances, with the distribution of labels being: 40% *contradiction*, 35% *entailment*, 25% *neutral*.

3.2 Set II: Temp-Duration

The motivation behind this dataset is to test whether language models can reason about fine-grained temporal durations. We frame this in an NLI format by having the premise mention an event occurring between two points in time, while the hypothesis mentions the same event having occurred for a given duration:

Premise : The war lasted **from 1939 to 1945**.

	Premise	Hypothesis	Label
a)	The meeting lasted *from 12 PM to 5 PM*.	The meeting lasted *for 5 hours*.	E
b)	The meeting lasted *from 12 PM to 5 PM*.	The meeting lasted *for 50 hours*.	C
c)	The meeting lasted *from 12 PM to 5 PM*.	The meeting lasted *for less than 5 hours*.	C
d)	The meeting lasted *from 12 PM to 5 PM*.	The meeting lasted *for less than 6 hours*.	E
e)	The meeting *began at 12 PM* and *lasted until 5 PM*.	The meeting lasted *for 5 hours*.	E
f)	The meeting lasted *from 9 PM to 3 AM*.	The meeting lasted *for 6 hours*.	E
g)	The meeting lasted *from 12 PM to 17:00*.	The meeting lasted *for 5 hours*.	E
h)	The spring quarter lasts *from Mar to June*.	The spring quarter lasts *for 3 months*.	E
i)	The war lasted *from July 1914 to Nov 1918*.	The war lasted *for 4 years 4 months*.	E
j)	The war lasted *from July 1914 to Nov 1918*.	The war lasted *for 52 months*.	E

Table 3: Variations in NLI pairs for duration calculation (E → entailment, C → contradiction). *a) - d)* are a few examples from the 6 basic pairs; *e)* is with a changed premise structure; *f)* is when the hypothesis time instance crosses over to the next cycle; *g), h)* are examples of choosing time instances from two different lists; *i), j)* are generation of specific dates using months and years in two different formats.

Hypothesis : The war lasted **for 6 years**.
Label : Entailment

We begin forming a basic NLI pair by choosing a sentence template. Based on the event's *temporal duration* label, a list from Table 1 is selected, and two time instances are randomly sampled without replacement. The smaller instance is mentioned in the premise as the event start time and the other instance as the event end time. We construct multiple hypotheses for the same premise. First, we calculate the *gold duration (GOLD)* by finding the difference between the two instances, assuming both the instances are part of the same cycle. Then, the hypothesis mentions the event to have occurred in two different settings *(equal to, less than)* for three different durations *(GOLD, GOLD+1, GOLD*10)*, generating a total of six hypotheses (Table 3 *a-d* are a few examples). We do this to test whether the NLI models can reason for the claimed duration's validity only when they are very distant *(GOLD*10)* or also very close *(GOLD+1)* to the gold duration. Generation of true labels for the pairs is automated, producing an *entailment* or *contradiction* depending on whether the gold duration falls in the duration range specified by the hypothesis.

We again introduce two variations in the dataset creation process. First, we change the wording of the premise sentence (Table 3 *e*). Secondly, while sampling the time instances, we force the ending instance to be picked such that it falls before the starting instance in the list, which implies that the event crossed over to the next cycle. In such cases, the calculation of the gold duration is slightly different (Table 3 *f*). We perform this next cycle calculation for all lists in Table 1 except *month-days* (because gold calculation without specifying the exact month is ambiguous) and *years* (because the list is acyclic). We also allow a similar blend of temporal expressions, like in *Temp-Order* set, combining the two *hours* (Table 3 *g*) and *months* (Table 3 *h*) lists. We construct specific dates by including *months* and *years* and allow the duration to be mentioned in a year-month format (Table 3 *i*) or a months-only format (Table 3 *j*).

We create separate train and test datasets using the same split of 53 and 18 templates as before. For each template, time instances are chosen based on their *temporal duration* label, along with the variations as mentioned above applied (Table 3 *e-j*). Running each template for five iterations produces a train dataset of 13,500 instances and test dataset of 3,540 instances, with the label distribution: 50% *entailment* and 50% *contradiction*.

3.3 Set III: Cross-Unit Duration

The motivation behind the creation of the *Cross-Unit Duration* set is to test whether language models understand the conversion relationship between magnitudes specified in different units of time; for instance, if models are able to interpret that 5 hours are less than 350 minutes but more than 250. Moreover, we investigate if these models are better at

	Premise	Hypothesis	Label
a)	The store will close *in 2 hours*.	The store will close *before 40 minutes*.	C
b)	*In 2 hours*, the store will close.	The store will close *after 84 minutes*.	E
c)	The store will close *in 2 days*.	*After 34 hours*, the store will close.	E
d)	*After 4 days*, the store will close.	The store will close *before 38 hours*.	C
e)	The store will close *before 4 days*.	*Before 174 hours*, the store will close.	E
f)	The store will close *before 6 hours*.	The store will close *after 77 minutes*.	N
g)	*After 3 hours*, the store will close.	The store will close *after 409 minutes*.	N

Table 4: Variations in NLI pairs for cross-unit duration comparison (E → entailment, C → contradiction, N → neutral). *a)* is the basic pair; *b), c)* are variations of basic pair with template position changed; *d) - g)* are variations in which the premise event occurs over a range of time.

certain kinds of conversions. We frame this task in an NLI format in a similar manner to the *Temp-Order* set. In the premise, we mention a future event that will occur after a given duration *(T1)*, while in the hypothesis we mention the same future event to occur before or after a different duration *(T2)*. Apart from varying magnitudes, *T1* and *T2* are also specified in different but adjacent units of time. More specifically, *T1* is specified in the higher adjacent unit of time, i.e., if *T2* is specified in minutes, then *T1* will be specified in hours. Since the premise mentions an event occurring at a future point in time while the hypothesis mention an event occurring over a time interval bounded on just one side, the premise event either lies in the interval or not, leading to the labels *entailment* and *contradiction* respectively. We tried multiple variations similar to *Temp-Order* set, like changing the position of the template in the premise/hypothesis (Table 4 *b, c*) and making the event in the premise also occur over a future interval (Table 4 *d-g*). The labeling procedure of the second variation is again similar to *Temp-Order* set.

To create the challenge set, we first pick a template and look at the list of its *temporal occurrence* values. We then iterate over all adjacent values in this list, e.g., seconds-minutes, hours-days, months-years. For each pair, we iterate over a manually created list of magnitudes for the higher unit of time *(T1)* for the premise. We then pick a magnitude in the smaller unit of time *(T2)* which is either higher or lower than *T1*. *T2*'s value is generated randomly, but a *difference range* parameter controls its absolute difference with *T1*'s value. For each fixed template, fixed duration unit pair, and fixed magnitude of *T1*, we generate twelve different premise-hypothesis pairs, four in which the premise occurs at a future point in time, and eight in which it occurs over a future interval of time. Using a *difference range* parameter of 5, we create a training set of 42,240 rows and a test set of 15,840 rows. Due to the challenge set creation procedure, the resultant dataset is naturally balanced with the same number of samples for each of the three labels.

4 Experimental Setup

We evaluate three different NLI models on each of our challenge sets. The first model is a pre-trained RoBERTa-large model fine-tuned on the MNLI corpus, which reports 90.8% accuracy on the MNLI-matched task. The second model is Microsoft's DeBERTa-large model fine-tuned on the MNLI corpus, which reports 91.9% accuracy on the MNLI-matched task. Both these models are trained on all three labels: *entailment*, *contradiction*, and *neutral*.

The third model comes from Vashishtha et al. (2020), which is a RoBERTa-large model fine-tuned on their temporal NLI datasets. For *Temp-Order* set, we evaluate their model trained on the *UDS-NLI (order)* corpus as it explored ordering relations between events, and we wanted to analyze if any of that knowledge transfers over for determining the ordering between temporal expressions. Similarly, for *Temp-Duration* and *Cross-Unit Duration* sets, we evaluate their model trained on the *UDS-NLI (duration)* corpus, which explored more coarse-grained duration of events. In contrast to the models fine-tuned on MNLI, these models are only trained on binary classification - producing *'entailed'* for the *entailment* label and *'not-entailed'* for the *contradiction* and *neutral* labels. We have a separate majority baseline corresponding to these models.

We report performances of all datasets under

Model	Method	Accuracy	F1 Score
Majority	Ternary Classification	40.29	23.14
	Binary Classification	65.19	51.46
RoBERTa (MNLI)	Direct Evaluation	52.75	45.36
	Hypothesis Only Training	40.29 $\pm$ 0	23.14 $\pm$ 0
	Train and Evaluate	99.81 $\pm$ 0.03	99.81 $\pm$ 0.03
DeBERTa (MNLI)	Direct Evaluation	51.57	44.29
	Hypothesis Only Training	40.29 $\pm$ 0	23.14 $\pm$ 0
	Train and Evaluate	99.76 $\pm$ 0.04	99.76 $\pm$ 0.04
UDS-NLI (order)	Direct Evaluation	56.36	57.20

Table 5: Evaluating *Temp-Order* set on NLI models

three different settings:

1. **Direct Evaluation**: Evaluating the pre-trained NLI models directly on the test splits of our challenge sets.

2. **Train and Evaluate**: Fine-tuning the NLI models with the train splits and reporting performances on the test splits. We report this to recognize the complexity of the synthetic datasets, and the ceiling performances that various NLI models can achieve on them.

3. **Hypothesis Only Training**: Fine-tuning the NLI models in a hypothesis-only setting (Poliak et al., 2018) with the train splits, and reporting performances on the test splits. We report this as a control for the results achieved in the *Train and Evaluate* setting.

We do not train the models fine-tuned on UDS-NLI corpora, and only report their performance under the *Direct Evaluation* setting, as the architecture of those models is similar to that of the pre-trained RoBERTa-MNLI model and we hypothesize that this may lead to similar results on training. More details on the training process are mentioned in Appendix A.

5 Results & Discussions

We present the results of all three challenge sets separately.

5.1 Set I: Temp-Order

Results of *Temp-Order* set are summarised in Table 5. When evaluating on the RoBERTa and DeBERTa models, there is an improvement of about 10% over the majority baseline. On analyzing the effect of different variations mentioned in Table 2, we find that changing the template position or its tense does not produce any significant difference in performance. However, we find that pairs where the premise event occurred at a fixed time instance (2 *a*) have an average of 75% accuracy, while the pairs where the premise event occurred over a time interval (2 *d-f*) have an average accuracy of 28%. This implies that models trained on MNLI have some basic understanding of temporal ordering, especially in determining whether a fixed time instance is present in another time interval. However, it gets difficult to reason about the ordering between two time intervals, where discerning the label is also not as straightforward.

We further analyze the accuracies for different methods of choosing the two time instances, and the results for DeBERTa are summed up in Figure 1. For pairs where the premise event takes place at a fixed point in time, most methods in which both time instances were sampled from the same list give over 73% accuracy. Among the methods in which time instances are sampled across multiple lists, dates in which *months* are combined with *years* give an average accuracy of 74%, but this drops to 57% when *month-days* are added, signifying that the comparison of specific dates becomes too complicated for the NLI model. The model also has a hard time mapping *hours* between 12 and 24-hour format, giving only 59% accuracy. For pairs where the premise takes place over a time interval, the accuracies of all methods are below 35%.

We also evaluate the RoBERTa model trained on *UDS-NLI (order)* corpus on our challenge set. However, the average performance was only 56%, even below the majority baseline for the corpus, not indicating any significant transfer of knowledge from their task of event-based temporal ordering.

The hypothesis-only baseline for both the RoBERTa/DeBERTa models is exactly the major-

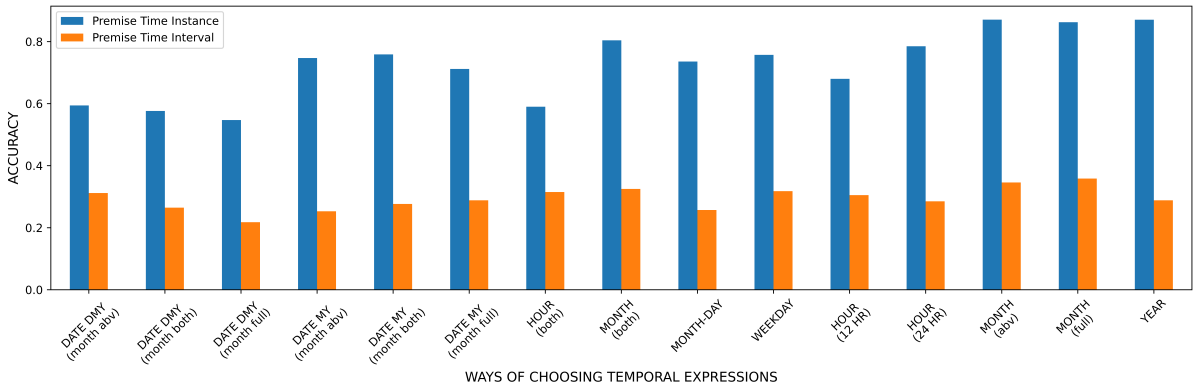

Figure 1: Comparison of accuracies across different ways of choosing temporal expressions when running *Temp-Order* set on DeBERTa fine-tuned on MNLI. Label 'DATE DMY' implies generating dates using all *month-day*, *month* and *year* (Table 2 *j*), while 'DATE MY' uses only *month* and *year* (Table 2 *i*). The added descriptions signify where the month instances are drawn from, where 'month (full)' implies that both months come from the *month (full name)* list, 'month (abv)' implies that they come from *month (abbreviated)* list, and 'month (both)' implies that one month instance comes from each of the two lists. Labels 'MONTH (both)' and 'HOUR (both)' signify similar methods of choosing from multiple lists (Table 2 *h* and *g* correspondingly).

ity baseline, which is not surprising as the true label cannot be determined without knowing the premise time instance. Further fine-tuning the pre-trained MNLI models on our *Temp-Order* set leads to an almost perfect accuracy of 99%. This is despite having separate templates for the train and test split, and having time instances randomly sampled from different lists. Since our method of generating labels was automated and depended on the values of the time instances, we infer that the numerous parameters of a large language model were able to learn this label generation process from the artificial NLI data.

5.2 Set II: Temp-Duration

Results for *Temp-Duration* set are summarized in Table 6. Both the RoBERTa and DeBERTa models fine-tuned on MNLI produce poor accuracies of around 55%, just over the majority baseline. While analyzing the DeBERTa predictions, we found that the model produced *entailment* for 83% of the data points, implying that it is not able to adequately determine durations. The different variations (Table 3) or methods of sampling time instances also did not have any significant effect. We investigated the performances of the six types of hypotheses and found that among the hypothesis types with the *contradiction* gold label, *'equal to GOLD*10'* produced 0.68 F1-score, compared to *'equal to GOLD+1'* and *'less than GOLD'*, which produced 0.15 and 0.07 F1-scores respectively. This might indicate that while the NLI model has difficulty figuring out when the claimed duration is incorrect, it still does better off when it is very distant *(GOLD*10)* from the gold duration compared to when it is very close *(GOLD+1)*. We also find that for the *'equal to GOLD'* hypothesis, pairs of instances far apart in a list tend to be misclassified more than pairs of instances that are closer to each other. This implies that determination of exact duration gets difficult for the NLI model as the distance between instances increases.

We also evaluate the model trained on *UDS-NLI (duration)* corpus on our set, and the results are slightly above the majority baseline. While the predictions by this model were not as skewed, we could not find any significant impact of the variations or the different methods of sampling time instances, not indicating any possible knowledge transfer from their problem of determining coarse-grained event duration.

On fine-tuning our challenge set under the hypothesis-only setting, both MNLI models surprisingly produce at least 15% gains in accuracy over the majority baseline. We investigate and find that the models use lexical cues from the different hypothesis types, producing *entailment* for all *'less than'* hypotheses. For the *'equal'* hypotheses, they predict *contradiction* when the claimed duration is a large value (more likely to be *GOLD*10*) and *entailment* when it is smaller (more likely to be *GOLD*). However, under the standard NLI training

Model	Method	Accuracy	F1 Score
Majority[1]	Binary Classification	50.00	33.33
RoBERTa (MNLI)	Direct Evaluation	54.32	46.64
	Hypothesis Only Training	68.82 ± 0.22	66.84 ± 0.51
	Train and Evaluate	91.86 ± 6.42	91.85 ± 6.42
DeBERTa (MNLI)	Direct Evaluation	56.67	51.53
	Hypothesis Only Training	64.15 ± 0.27	59.64 ± 0.41
	Train and Evaluate	73.72 ± 3.92	73.28 ± 4.50
UDS-NLI (duration)	Direct Evaluation	58.44	57.66

Table 6: Evaluating *Temp-Duration* set on NLI models

Model	Method	Accuracy	F1 Score
Majority	Ternary Classification	33.33	16.67
	Binary Classification	66.67	53.33
RoBERTa (MNLI)	Direct Evaluation	35.47	28.71
	Hypothesis Only Training	49.38 ± 0.71	39.51 ± 0.52
	Train and Evaluate	99.97 ± 0.02	99.97 ± 0.02
DeBERTa (MNLI)	Direct Evaluation	45.02	38.60
	Hypothesis Only Training	49.58 ± 0.79	41.29 ± 0.56
	Train and Evaluate	99.94 ± 0.03	99.94 ± 0.03
UDS-NLI (duration)	Direct Evaluation	52.61	53.54

Table 7: Evaluating *Cross-Unit Duration* set on NLI models

scenario, these cues are not the only factor behind learning, as the RoBERTa MNLI model produces 91.86% average accuracy, which is a gain of 20% over the hypothesis-only setting. Among the various methods of sampling time instances, *years* performs the worst, producing only 66% accuracy, possibly because the lengths of duration can be as large as 100 years. Finally, the hypothesis types *'equal to GOLD*10'* and *'less than GOLD*10'* produce 99% accuracy, while *'equal to GOLD'* and *'less than GOLD'* report below 90%, confirming our speculation that it is easier for the models to reason about the validity of the claimed duration when it is distant from the gold duration.

5.3 Set III: Cross-Unit Duration Set

As shown in Table 7, all the models produce poor performances on direct evaluation, just near the majority baseline. DeBERTa fine-tuned on MNLI manages to perform better when compared to RoBERTa fine-tuned on MNLI by around 10% on overall accuracy. Hence, we can conclude that DeBERTa has a slightly better understanding of cross-unit duration comparison when compared to RoBERTa.

[1]Same majority baseline because no neutral labels.

Similar to *Temp-Order* set, all models performed better compared to their respective majority baselines when the premise event occurs at a future point in time rather than over a time interval. More specifically, we see an improvement in accuracy of around 18% (29.65 to 47.12) for RoBERTa and around 10% (41.77 to 51.52) for DeBERTa when we switch from the premise occurring over a time interval to a point in time.

We analyzed the results on adjacent units to recognize if there are specific pairs for which the models are better able to figure out the conversion relationship. We did not find any significant pair for RoBERTa or DeBERTa models, but we find that the *UDS-NLI (duration)* model does better on bigger unit pairs of duration, i.e., it performs the best on conversion between month-years (56.03% F1), then day-months (55.32% F1), then hours-days (51.02% F1), and then minutes-hours (16.74% F1). This suggests a better transfer of knowledge for bigger time units from the *UDS-NLI (duration)* model to our challenge set.

Similar to *Temp-Duration* set, on fine-tuning under the hypothesis-only setting, both MNLI models produce around 16-17% gains in accuracy over the majority baseline using lexical cues present in the

hypotheses due to the challenge set creation process. For the hypotheses that contain *'before'*, the models tend to predict *entailment* if the duration *(T2)* is large, and *contradiction* if it is small. Similarly, for the hypotheses that contain *'after'*, the models mostly predict *contradiction* if the duration is large, and *entailment* if it is small. However, on standard training, the accuracy goes up from around 50% to near perfect 99%, showing that these cues are not the only reason behind the model's performance and that it actually learns the relationship between the premise and hypothesis.

We believe a valuable addition to this challenge set would be introducing more varied phrasing of prepositions. That is, using synonymous ways of denoting a temporal event occurring before, after, or strictly at a point in time. In particular, phrasing like 'after the next 60 minutes' or 'sometime after 60 minutes pass' could be examples of more specific ways to represent that an event occurred 'after 60 minutes' - a phrase which we acknowledge may read to mean 'in exactly 60 minutes', as opposed to some time after.

6 Conclusion

We create three challenge sets that test different kinds of relationships between temporal expressions. We evaluate these challenge sets on popular NLI models like RoBERTa and DeBERTa trained on MNLI, and find that while they can reason about simple cases of ordering between time instances, they fail when presented with more complicated cases or when temporal reasoning requires determining fine-grained duration. Since our challenge sets were synthetically created, training on them helped the NLI models to figure out the label generation process, and they produced near-perfect accuracy for the *Temp-Order* and the *Cross-Unit Duration* sets. A direction for future research could be evaluating and comparing models, trained on other NLI datasets containing temporal expressions, on our challenge sets. Another direction could be to collect naturally occurring sentences that contain temporal expressions from large corpora and recast them into NLI format for similar testing of understanding of temporal expressions.

References

Steven Bethard, Leon Derczynski, Guergana Savova, James Pustejovsky, and Marc Verhagen. 2015. SemEval-2015 task 6: Clinical TempEval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 806–814, Denver, Colorado. Association for Computational Linguistics.

Taylor Cassidy, Bill McDowell, Nathanael Chambers, and Steven Bethard. 2014. An annotation framework for dense event ordering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 501–506, Baltimore, Maryland. Association for Computational Linguistics.

Sanxing Chen, Guoxin Wang, and Börje Karlsson. 2019. Exploring word representations on time expression recognition. Technical report, Microsoft Research Asia.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Wentao Ding, Guanji Gao, Linfeng Shi, and Yuzhong Qu. 2019. A pattern-based approach to recognizing time expressions. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6335–6342.

Quang Do, Wei Lu, and Dan Roth. 2012. Joint inference for event timeline construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 677–687, Jeju Island, Korea. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention.

Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. 2014. Context-dependent semantic parsing for time expressions. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1437–1447, Baltimore, Maryland. Association for Computational Linguistics.

Artuur Leeuwenberg and Marie-Francine Moens. 2018. Temporal information extraction by predicting relative time-lines. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1237–1246, Brussels, Belgium. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Aakanksha Naik, Luke Breitfeller, and Carolyn Rose. 2019. TDDiscourse: A dataset for discourse-level temporal ordering of events. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 239–249, Stockholm, Sweden. Association for Computational Linguistics.

Qiang Ning, Hao Wu, Rujun Han, Nanyun Peng, Matt Gardner, and Dan Roth. 2020. Torque: A reading comprehension dataset of temporal ordering questions.

Qiang Ning, Ben Zhou, Zhili Feng, Haoruo Peng, and Dan Roth. 2018. CogCompTime: A tool for understanding time in natural language. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 72–77, Brussels, Belgium. Association for Computational Linguistics.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. 2018. Hypothesis only baselines in natural language inference. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 180–191, New Orleans, Louisiana. Association for Computational Linguistics.

James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003a. Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

James Pustejovsky, Patrick Hanks, Roser Saurí, Andrew See, Rob Gaizauskas, Andrea Setzer, Dragomir Radev, Beth Sundheim, David Day, Lisa Ferro, and Marcia Lazo. 2003b. The timebank corpus. *Proceedings of Corpus Linguistics*, page 40.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Jannik Strötgen and Michael Gertz. 2010. HeidelTime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324, Uppsala, Sweden. Association for Computational Linguistics.

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

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

Siddharth Vashishtha, Benjamin Van Durme, and Aaron Steven White. 2019. Fine-grained temporal relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2906–2919, Florence, Italy. Association for Computational Linguistics.

Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. 2007. SemEval-2007 task 15: TempEval temporal relation identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80, Prague, Czech Republic. Association for Computational Linguistics.

Marc Verhagen, Roser Saurí, Tommaso Caselli, and James Pustejovsky. 2010. SemEval-2010 task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, Uppsala, Sweden. Association for Computational Linguistics.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5307–5315, Hong Kong, China. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 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. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. "going on a vacation" takes longer than "going for a walk": A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3363–3369, Hong Kong, China. Association for Computational Linguistics.

A Training Details

We use three different NLI models for our experiments. For the RoBERTa and DeBERTa models fine-tuned on MNLI, we use the `roberta-large-mnli` and `microsoft/deberta-large-mnli` models respectively, available under the `transformers` library from HuggingFace (Wolf et al., 2020). For the *UDS-NLI* models, we directly use the saved RoBERTa-large models for *UDS-NLI (duration)* and *UDS-NLI (order)* made publicly available by Vashishtha et al. (2020).

For training, we use an Adam optimizer, with a learning rate of $2 * 10^{-5}$ and 0.1 weight decay. We use a batch size of 16 for training and 128 for testing. We train for 10 epochs, using early stopping with a patience of 2. We run each experiment for three random seeds (3, 5, 7), and use them to calculate the mean and standard deviation for the accuracy and F1 (weighted) score metrics.