Self-Supervised Logic Induction for Explainable Fuzzy Temporal Commonsense Reasoning

Bibo Cai[1], Xiao Ding[1]*, Zhouhao Sun[1], Bing Qin[1], Ting Liu[1], Baojun Wang[2], Lifeng Shang[2]

[1]Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
[2]Huawei Noah's Ark Lab
{bbcai, xding, zhsun, bqin, tliu}@ir.hit.edu.cn
{puking.w, Shang.Lifeng}@huawei.com

Abstract

Understanding temporal commonsense concepts, such as times of occurrence and durations, is crucial for event-centric language understanding. Reasoning about such temporal concepts in a complex context requires reasoning over both the stated context and the world knowledge that underlines it. A recent study shows massive pre-trained LM still struggle with such temporal reasoning under complex contexts (e.g., dialog) because they only implicitly encode the relevant contexts and fail to explicitly uncover the underlying logical compositions for complex inference, thus may not be robust enough. In this work, we propose to augment LMs with the temporal logic induction ability, which frames temporal reasoning by defining three modular components: temporal dependency inducer and temporal concept defuzzifier, and logic validator. The former two components disentangle the explicit/implicit dependency between temporal concepts across context (before, after, ...) and the specific meaning of *fuzzy* temporal concepts, respectively, while the validator combines the intermediate reasoning clues for robust contextual reasoning about the temporal concepts. Extensive experimental results on TIMEDIAL, a challenging dataset for temporal reasoning over dialog, show that our method, **L**ogic Induction **E**nhanced **C**ontextualized **TE**mporal **R**easoning (LECTER), can yield great improvements over the traditional language model for temporal reasoning.

Introduction

Understanding *time* in natural language text is crucial for understanding the evolving world. Humans can reason about temporal concepts in a language such as the times of occurrence and durations of events based on their rich temporal commonsense knowledge. Many event-centric NLP systems, such as timeline construction (Leeuwenberg and Moens 2018), clinical analysis (Bethard et al. 2015), dialogue assistants (Rong et al. 2017), etc, also rely on the ability of temporal commonsense reasoning to achieve satisfactory performance.

As manually annotating the Temporal CommonSense (TCS) knowledge required for temporal reasoning is time and labor-consuming, recent studies attempt to improve the NLP system's TCS reasoning ability with cheap supervision: the typical approach is to identify the temporal mentions in the free-form text, which is utilized to augment the traditional language model objective function with temporal signals. We note that, however, there are significant limitations with such a pretraining-only approach.

S1: *I have a meeting this afternoon.*
S2: *When will it begin?*
S1: *It will begin at three o'clock, what's the time now?*
S2: *It's _____.*
S1: *I have to go now, I do not want to be late!*

(a)

Passage: *Taunton has four art galleries... Hughes Gallery founded in 2007... Art Euphoric founed in 2008 has both visual and craft exhibits...*
Question: *How many years after founding of Hughes was Art Euphoric founed?*

(b)

Figure 1: Fuzzy Reasoning. Temporal reasoning should resolve the dependency among global temporal concepts and figure out the specific meaning of the fuzzy temporal concepts.

As shown in Figure 1 (a), to fill the correct answer in the blank, one should be able to reason about the global context to derive the latent inter-dependencies among the temporal concepts in the context (the target answer should be earlier than the temporal expression appears in the position of "three o'clock").

Moreover, different from the traditional tasks that also require such a relation induction ability, it also presents additional practical challenges. For example, for the QA problem in Figure 1 (b), the right answer can be derived once the correct computation logic is obtained (i.e., $2008 - 2007 = 1$). While in the example shown in Figure 1 (a), we should not only understand the latent dependencies (i.e., "`earlier`") among the temporal concepts, but also infer that the *fuzzy* expression "three o'clock" in the example denotes a time

*Corresponding author

of the day, most probably 3 pm. The target answer could be inferred from the logical constraint earlier("3 pm", ?). Hence, the temporal reasoning process may implicitly involve three steps: resolve the dependency among global temporal concepts and figure out the specific meaning of the fuzzy temporal concept, followed by utilizing them together with the general temporal knowledge to verify the correctness of the candidate temporal concept.

While the traditional pre-trained objectives only implicitly encode the relevant contexts and fail to explicitly reason with the underlying logic rules for temporal inference, which lead to reliance on shallow cues for complex reasoning (Helwe, Clavel, and Suchanek 2021; Qin et al. 2021).

To solve these problems, in this work, we present LECTER, a novel self-supervised framework that conducts temporal logic induction for fuzzy temporal commonsense reasoning. Specifically, LECTER frames temporal reasoning by introducing three modular components: the temporal dependency inducer, temporal concept defuzzifier, and logic validator. The dependency inducer disentangles the explicit/implicit dependency between temporal concepts across context (before, after, ...) and the defuzzifier resolves the specific meaning of *fuzzy* temporal concepts. The logic validator is a symbolic module defined with the logic programming language DeepProbLog (Brown et al. 2020)), which combines the resolved results of the former two modules to verify the correctness of the candidate temporal concepts. The clues provided by the verification results work together with the LM-based predictor for TCS reasoning. The LECTER is trained with two self-supervised objectives: the regression-based temporal value recovery objective and the temporal logical entailment objective. The former objective trains the model to learn the shallow temporal knowledge from context, while the latter objective aims to teach models to uncover the logical rules that could explain the reasoning process based on implication. The whole network is trained in an end-to-end manner.

We evaluate LECTER on the challenging contextual TCS reasoning dataset TIMEDIAL (Qin et al. 2021), which demonstrates significant performance gains (more than 10 points) across the LMs continually trained with traditional pre-training objectives. Furthermore, with the temporal logic induction module, LECTER could capture the underlying logic to make the decision, which makes it more explainable than common language model-based methods.

Background

Problem Definition

The document D contains sequences $[s_1, s_2, \cdots, s_M]$ and temporal expressions $t_i \in \Gamma, 1 \leq i \leq N$, where M and N are the total number of sequences and time expressions in D with $N > 1$. Γ represents the set of temporal expressions. Formally, the task is formulated as a *clozen* task: given the corrupted document D' where a temporal expression in D is masked out, a system is required to select all suitable temporal expressions for the masked-out span from the answer set $\mathcal{A}$.

For example, both "two o'clock" and "2:00 pm" are correct answers for the blank in the passage shown in Figure 1 (a), while "8:00 am" is wrong.

The DeepProbLog Framework

We notice that human's prior knowledge of the numerical relation between temporal concepts can be utilized as distant supervision for the temporal logic induction, which can be effectively integrated into the neural network with the neural-symbolic framework DeepProbLog. Before diving into the details of utilizing DeepProbLog to describe our problem, we first provide a basic overview of the DeepProbLog (see (Manhaeve et al. 2021) for more details), and illustrate how the background knowledge is integrated.

Generally, the DeepProbLog is a *probabilistic logic programming* language that incorporates deep learning by means of *neural predicates*.

Definition 1 (**Probabilistic Logic Programming**). The probabilistic logic programming (Raedt, Kimmig, and Toivonen 2007) is a programming paradigm that is largely based on formal logic. A program written with probabilistic logic programming language contains a set of probabilistic facts $\mathcal{F}$ of the form $p :: f$ where p is a probability and f an atom, and a set of rules $\mathcal{R}$.

Example. The following program defines the domain of tossing coins.

```
1 0.4 :: coin(x1,h).
2 0.5 :: coin(x2,h).
3 twoHeads(X,Y) :- coin(X, h), coin(Y, h).
```

Here, coin(x_1, h) denotes the fact that the coin x_1 lands on the head, which is an atom with the predicate coin and two arguments. The rule twoHeads(X, Y) :- coin(X, h), coin(Y, h) defines what it means for both coins to land on heads, which is in the form of h :- $b_1, \cdots, b_n$ denoting the logical implication: "h if b_1 and $\cdots$ and b_n"

The main inference task in logic programming is to compute the true probability of the *query* atom q in the canonical model of the logic program P. In this case, given the query twoHeads(x_1, x_2), the program can obtain the probability of both coins landing on heads, which is the product of both probabilities: P(twoHeads(x_1, x_2))=0.2.

Definition 2 (**Neural Predicate**). A neural predicate in DeepProbLog is the predicate that allows instantiating probabilistic facts whose probabilities are parameterized by neural networks processing raw data.

To declare neural predicates, DeepProbLog enhances the traditional logic programming language with a primitive for neural extension:

$$nn(n_{id}, \mathbf{x}, z, \mathcal{Z})$$

where nn is a reserved functor used to declare a neural predicate, n_{id} is an identifier for the underlying neural network, $\mathbf{x}$ and z are the input raw data, and the output symbolic label of the neural network and $\mathcal{Z}$ denotes the domain of the output distribution by the neural network.

Example. The following DeepProbLog program defines the neural extension of the domain of tossing coins.

```
1 nn(coin_nn, [X], S, [h,t]): coin(X,S)
2 twoHeads(X,Y) :- coin(X, h), coin(Y, h).
```

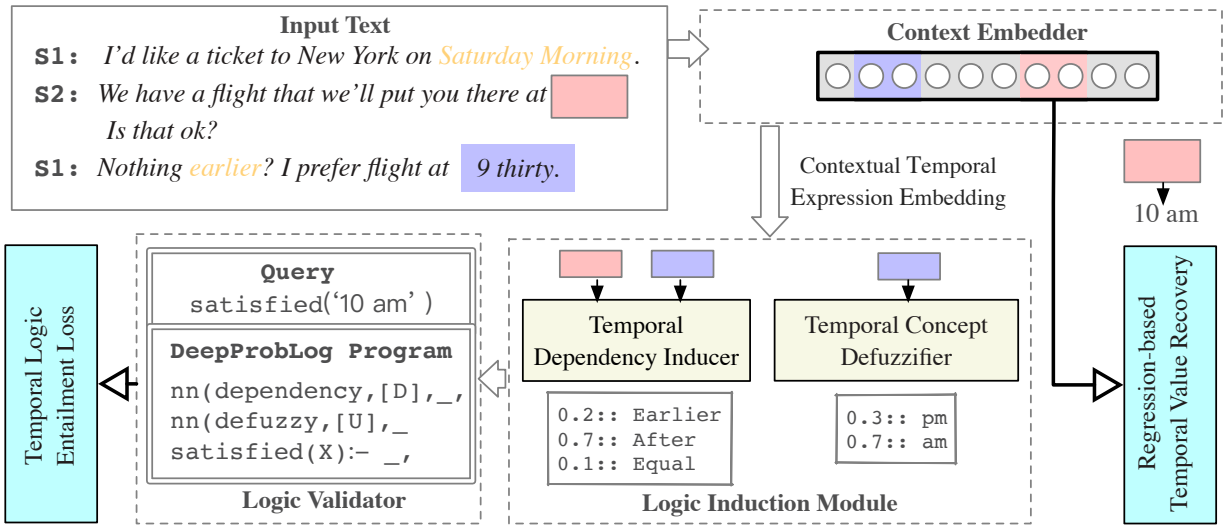

Figure 2: The framework of LECTER. We leverage the logic induction module to resolve the intermediate inference steps for fuzzy temporal commonsense reasoning. The logic validator acquires the predicted probability distributions from the dependency inducer and concept defuzzifier to compute the temporal logic entailment loss. It works together with the regression-based temporal value recovery loss to train the model in an end-to-end manner.

In this case, the `coin` is a neural predicate. The neural network represents a discriminative classifier, which maps the raw input X (can be an image of the tossing results) to the distribution over $[h, t]$ (head and tail). The probability of the facts with predicate *coin* is computed by the softmax layer of the neural network.

The training of these neural predicates is done by providing supervision on the head of the logical rules expressed as standard logical queries.

Definition 3 (**Learn From Entailment**). Given a DeepProbLog program with parameters Θ, the inputs $\mathcal{X}$ and the query q that is desired to be true, the model adjusts the weights to maximize query probabilities $P_\Theta(q|\mathcal{X})$ for all training examples.

If the average negative log-likelihood is chosen as the loss function, the loss is in the form of:

$$\arg\min_{\Theta} \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} -\log P_\Theta(q|\mathcal{X}) \tag{1}$$

With the help of aProbLog, the gradient of the loss can be effectively computed which is then used in standard gradient-based methods for optimizing parameters in an end-to-end manner.

Method

In this work, we augment the large-scale language models with temporal logic induction ability for fuzzy temporal commonsense reasoning. Our method, LECTER (depicted in Figure 2), consists of three components: (i) a context embedder that encodes the input context rich in temporal concepts. (ii) the logic induction module composed of the *logic inducer* and the *temporal concept defuzzifier*, which aims to predict the distribution of the context-dependent temporal relation among global temporal concepts, and the specific meaning of a fuzzy temporal concept. (iii) the logic validator which encodes human's prior knowledge about the numerical relation between temporal concepts by the neural logic programming language DeepProbLog, allowing to estimate the degree that a query temporal concepts are consistent with the output of the neural layers. For effective training of the LECTER model, we propose two self-supervised learning objectives, i.e., the *Regression-based Temporal Value Recovery* and *Temporal Logical Entailment*.

Context Embedder

LM-Based Encoder The LM-based encoder learns an initial representation of the input tokens with an MLM-based pre-trained Transformer encoder (e.g., BERT (Devlin et al. 2019), RoBERTa (Liu et al. 2019)). The self-supervised input data is constructed in two steps. First, we utilize the temporal expression identifier to identify and normalize the temporal concepts in the context C. Second, for each input context, a single temporal expression that contains numerals will be masked out resulting in a cloze test x (i.e., the input data). We denote the temporal mention to be recovered as *target concept*, the other temporal expressions in the same context are denoted as *auxiliary concepts*.

Temporal Expression Identification Aiming at automatically identifying temporal concepts from texts, we utilize the off-line temporal expression extraction tool (Chang and Manning 2012) together with heuristic rules to construct a grounded representation of the temporal mention. As shown in Figure 3, the temporal mention is grounded as a tuple $TE =< M, N, U, D >$, where M is a collection of the tokens of temporal mention, N and U are the numeric and

raw context:

… The meeting will be started at 8 in the morning.

normalization:

```
M -> "8 in the morning"
N-> "8"
U -> "am"
T -> "Clock"
```

Figure 3: The temporal concept parser parses the temporal expressions in context into the normalized format as a tuple.

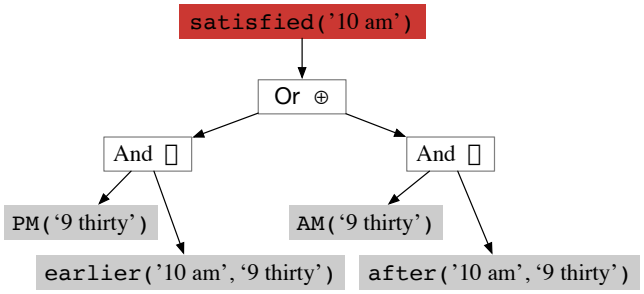

Figure 4: The probability forwarding process of the logic validator.

unit of the mention. D represents the temporal dimension (e.g., CLOCK, DURATION). Note, the unit U may not always be expressed explicitly in the text, which leads to a TE with N = `None`. For example, the specific clock time of the mention "at eight o'clock" needs to be further resolved (also known as fused head resolution (Elazar and Goldberg 2019). We call such kind of temporal concepts as *fuzzy* temporal concept. The origin temporal mention in the context will be replaced with its normalized version, which consists of two tokens: `value` and `unit` (e.g., "3 o'clock in the afternoon" → "3 pm"). For the fuzzy temporal mention, we take the `[MASK]` as the `unit` token.

Language Encoding Given a target temporal concept t in the input data x, we mask out t from x to construct the sequence of input tokens $x_{/\{t\}} = \{w_{int}, w_1, w_2, \cdots, w_T\}$, then compute the contextualized representation $\{\boldsymbol{h}_{int}, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_T\}$ for each token with pre-trained language model. Here, $\boldsymbol{h}_t$ denotes the contextualized representation of the t-th token obtained with the LM, $x_{/\{e\}}$ denotes replacing x with *two* mask token `[MASK]s`. The intuition is that the two `[MASK]s` are corresponding to the numeral and unit part of the temporal concept, respectively.

Logic Induction Layer

Temporal Dependency Inducer The temporal dependency inducer aims to generate the context-dependent temporal logic that the target and auxiliary temporal concepts should follow. In this paper, we consider logic rules indicating the binary latent relationship, so the induction can be formulated as a relation extraction task. Three kinds of relationships are taken into consideration: before, after, and equal.

Specifically, the temporal logic inducer generates the distribution of the relation between target t and auxiliary temporal concept a by:

$$\mathbf{r} = \mathbf{W}_{rule}[\mathbf{x}_t \parallel \mathbf{x}_a \parallel \mathbf{x}_t \odot \mathbf{x}_a]. \tag{2}$$

where $\mathbf{r}$ is the output probabilistic logits across the rule labels, $\mathbf{W}_{rule}$ is the learnable parameter matrix, the $\parallel$ denotes the tensor concatenation, $\mathbf{x}_t$/$\mathbf{x}_a$ denotes the representation of the target/auxiliary temporal concept, which is obtained by averaging over the contextualized embedding $\boldsymbol{h}_t$ of all the tokens w_t that make up the concept.

Temporal Concept Defuzzifier The uncertainty of fuzzy concepts makes it unable to directly instantiate the temporal logic rules generated by the logic inducer. To tackle this problem, we introduce the temporal concept defuzzifier module to estimate the probability of different normalization results of the fuzzy temporal concepts. As illustrated in the logic inducer, we obtain the representation $\mathbf{x}$ of temporal mention by averaging over the embedding of all the tokens that make up it. Specifically, each $\mathbf{x}$ is fed to the dimension-specific unit classification layer $\mathbf{U}_{dim}$ based on the dimension of the temporal mention:

$$\mathbf{u}_i = \mathbf{U}_{dim}\mathbf{x} \tag{3}$$

where $\mathbf{u}_i$ is the predicated logits over the label of the unit set in the given dimension, $dim \in \{\texttt{CLOCK}, \texttt{DURATION}\}$.

Logic Validator

The temporal logic rules generated in the previous step are further evaluated by instantiating the variables with relevant temporal concepts in the context. We aim to compute the true probabilities of the grounded rule (i.e., the true probability of the proposition "The target temporal concept is satisfied by the logic rules"), known as learning from entailment which will be maximized during training.

Specifically, the DeepProblog (Manhaeve et al. 2021) is a neural probabilistic logic programming language that integrates expressive probabilistic-logical modeling and reasoning with the neural networks. The idea is to take the outputs of both the logic rule inducer and temporal concept defuzzifier module as neural annotated disjunctions, once the symbolic grounding structure is given, the DeepProbLog program allows us to train the neural network seamlessly with back-propagation. An example of probability forwarding process can be seen in Figure 4.

Training

For effective training of the LECTER model, we propose two self-supervised learning objectives, i.e., the *Regression-basd Temporal Value Recovery* and *Temporal Logical Entailment*. The two losses work together to train the LECTER model.

Regression-Based Temporal Value Recovery Similar to many other self-supervised tasks such as masked language

modeling (Yang et al. 2020; Zhou et al. 2020, 2022a), we aim to teach models to recover the original temporal semantics of sentences from corrupted inputs. The regression-based temporal value recovery objective requires the LECTER model to predict the normalized temporal value given the context, where the original temporal concept tokens are removed. Previous work (Yang et al. 2020) shows that such an exact value prediction paradigm is very effective for duration prediction, in this work, we extend the idea for temporal reasoning in a more general domain.

Specifically, considering the final hidden vectors of the two mask tokens related to the explicit temporal concept e_i as $\mathbf{h}_i^{unit}$ and $\mathbf{h}_i^{num}$, they are fed to the dimension-specific regression layer $\mathbf{w}_{dim}$ based on the dimension of e_i:

$$v_i = \frac{1}{2}\mathbf{w}_{dim}(\mathbf{h}_i^{unit} + \mathbf{h}_i^{num}) \tag{4}$$

where v_i is the predicated normalized value, $dim \in \{\texttt{CLOCK}, \texttt{DURATION}\}$, as we consider the TE of CLOCK and the DURATION dimension.

Each v_i will be optimized by mean square error loss:

$$\mathcal{L}_{reg} = \sum_{i=1}^{q}(v_i - y_i)^2. \tag{5}$$

where y_i denotes the normalized temporal value of the temporal concept e_i. To construct y_i, we normalize the temporal mention in DURATION dimension to logarithmic "second" space (e.g., 2 hours $\rightarrow$ 7200 seconds $\rightarrow \log(7200) \rightarrow 8.9$). As for the temporal mention in CLOCK dimension, we represent time in decimal hours format (e.g., 7:30 pm $\rightarrow 19 + 30/60 \rightarrow 19.5$).

Temporal Logical Entailment Following DeepProbLog (Manhaeve et al. 2021), we utilize the "learning from entailment" loss function. Specifically, given the training examples $\mathcal{X}$ and q as the query, the model needs to adjust the weights to maximize query probabilities $P_\theta(q|\mathcal{X})$ for all training examples. This can be reached by minimizing the average negative log-likelihood of the query:

$$\mathcal{L}_{logic} = \sum_{q} -log(P_\theta(q|\mathcal{X})) \tag{6}$$

Such loss can be optimized with gradient-based learning which allows for seamless integration with neural training.

Consequently, the learning objective of LECTER is to minimize:

$$\mathcal{L} = \mathcal{L}_{reg} + \lambda * \mathcal{L}_{logic}. \tag{7}$$

At inference time, we predict the most plausible answer with:

$$\arg\min_{a \in \mathcal{A}} \mathcal{L}(a|D') \tag{8}$$

where A is the answer set and D' is the document with one temporal span masked out.

Experiment

Dataset

We evaluate the performance of our proposed LECTER model on the challenge dataset TIMEDIAL (Qin et al. 2021). TIMEDIAL is a temporal commonsense reasoning benchmark on the dialog. Given a multi-turn dialog where a span of temporal words is masked out, the task requires the model to predict the suitable substitutions for the span from 4 options, two of which are right and two of which are wrong. In TIMEDIAL, the dialogs are carefully curated and are rich in temporal concepts, hence the model should accurately understand the causal relations between the temporal concepts to make an accurate prediction. There are 1.1k test instances in total and each dialog contains 11.7 turns and 3 temporal concepts on average.

Evaluation Metrics

Following (Qin et al. 2021), we utilize the *2-best accuracy* metric to evaluate the model's performance in our experiment, which measures whether both of the model's top-ranked answers are correct.

Experiment Settings

To evaluate whether the model could learn transferable reasoning skills, we focus on an out-of-domain training setting: the model is pre-trained with a large-scale corpus from a general domain with the *self-supervised* objective and is evaluated on the TIMEDIAL dataset in a *zero-shot* manner. This training setting could also avoid the possible problem of data leakage that can occur when training models based on in-domain data. As the out-domain finetuning dataset used in (Qin et al. 2021) is not publicly available, in this work, we leverage other large-scale publicly available corpus containing over 700MB of text[1] to construct our self-supervised training dataset. The temporal expressions in the unsupervised dataset are identified with the temporal concept identifier as described in the previous section. After preprocessing, we obtain 97k/24k instances for training/validation.

We experiment with the base model of BERT (Devlin et al. 2019) and RoBERTa (Liu et al. 2019) to construct our contextual encoder. During the training, the batch size is set to 32. The combination weight λ in Eq.7 is set to 1. We search the learning rate with grid search in $lr \in \{5e-6, 1e-5, 5e-5\}$ for the baseline and LECTER. The implementation is based on Pytorch and trained on a Tesla V100 GPU with Adam optimizer with 10 epochs. Note, a small number of answers can not be scored with Eq.8 at inference time. We handle this problem by utilizing the traditional masked language model loss to score such answers.

Baseline

Following (Qin et al. 2021), we compare our approach with popular pre-trained language models with different modeling paradigms. For BERT and RoBERTa, the candidate temporal span with k tokens in the dialog is replaced with k mask tokens. The average of the mask recovery cross-entropy loss for each mask token is utilized as the prediction score. For T5, given a masked dialog context, we evaluate the likelihood of generating the given temporal span (normalized with the token number of the span). We report the

[1] https://github.com/qywu/DialogCorpus

Model	2-**best Acc(%)**
T5-base	
ZERO (Qin et al. 2021)	39.1
OUT (Qin et al. 2021)	51.9
BERT-base	
ZERO (Qin et al. 2021)	44.8
OUT (Qin et al. 2021)	53.7
LECTER	**65.8**
RoBERTa-base	
ZERO (Qin et al. 2021)	52.2
OUT (Qin et al. 2021)	59.3
LECTER	**71.5**

Table 1: Performance of TCS reasoning on TIMEDIAL dataset. Results show that LECTER outperforms basic PLMs by a large margin. T5 is not applicable to be utilized as the context encoder for LECTER.

PLM's performance both in a zero-shot manner (denoted as `ZERO`) and with additional fine-tuning under the same out-domain training dataset (denoted as `OUT`) as LECTER.

Experimental Results

Table 1 shows our main results. As we can observe, (1) with a temporal-aware continual pre-training, the performance of PLM can be improved by a large margin than the zero setting. This proves that temporal reasoning can benefit from the temporal signals acquired with cheap supervision. (2) Our model LECTER, enhanced by explicitly modeling the fuzzy temporal reasoning process, achieves the best performance on the 2-best accuracy score. Compared with the baseline models, the improvements are over 10% on both the BERT and RoBERTa models continually trained with the same out-domain data. This shows LECTER is much more effective in learning transferable temporal reasoning skills than plain PLMs.

Ablation Study

We develop an ablation study to test different variations of LECTER (take RoBERTa as the backbone), as shown in Table 2. We can see that (1) the regression-based temporal value recovery objective lifts the performance by 9.0%. This is not unexpected due to the *fuzziness* of temporal expression in language. For example, the general commonsense of the duration of "go to the office" could be "20 minutes", "half an hour", etc. As a result, the model should fully learn the numerical property of the temporal concepts. (i.e., "five days" has a similar range to "one week", "11:30 am" is earlier than "1:00 pm", etc.). Leveraging the off-line normalization module for temporal concepts and pre-training the model with a regression objective make the model naturally learn a continual representation of the contextual temporal commonsense. While the common language model pre-training objective deals with the temporal concept token-by-token and thus fails to capture the fuzziness of temporal expression. (2) the temporal logic induction module lifts the results by 2.1%. It demonstrates the importance of explicitly uncovering the underlying relational compositions for complex inference.

Model	**2-best Acc(%)**
LECTER	71.5
-Temporal Value Regression	62.5
-Temporal Logic Induction	69.4

Table 2: Ablation results. Results show significant performance lifts from both modules.

A temporal reasoning model has to understand the underlying reasoning process to achieve better generalization performance.

Case Study and Error Analysis

Table 3 shows the results of a case study with the outputs of LECTER and RoBERTa-OUT model (as illustrated in section). In the three cases, the first two answers of the four candidates are correct. In the first case, both the incorrect answers already partially occur in the context. The RoBERTa-OUT model was completely confused and rank both the incorrect answers as the top prediction for the blank, although the two options violate the context. This shows it may rely on shallow text matching for temporal reasoning. Similarly, in the second example, the RoBERTa-OUT model still fails to capture the underlying logical relation among the temporal concepts in context. Instead, our model explicitly seeks to uncover the logic for the observations based on implication, which makes it able to exclude the incorrect answer that violates the inducted logic rules. In case 3, the resolved logical constraint for reasoning about the target temporal expression should be "`after`(10 am, ?)", which is satisfied by the first three candidate answers. To obtain the correct answer, the model also requires knowing the usual time of occurrence of the event "have dinner". In such special cases that require additional temporal comprehension, LECTER may make a mistake.

Related Work

Temporal Commonsense Reasoning

The research topic of understanding *time* in natural language has long been studied for decades in the NLP community. A line of work related to temporal analysis focuses on the temporal information extraction task: temporal expressions extraction and normalization (Chang and Manning 2012; Lange et al. 2020; Ning et al. 2022; Cai et al. 2022), temporal relation among events extraction (Vashishtha, Van Durme, and White 2019; Han, Zhou, and Peng 2020; Han et al. 2021), and timeline construction (Viani et al. 2019).

Recently, multiple works have been done on the *Temporal CommonSense (TCS) reasoning* (Thukral, Kukreja, and Kavouras 2021; Zhou et al. 2022a,b), such as events' TCS property prediction (Zhou et al. 2020) (the duration, frequency, typical time of a sentence-level event), event ordering (Han, Ren, and Peng 2021) (how events are temporally arranged), and script learning (Lee and Goldwasser

Dialogue	Options	RoBERTa-OUT	LECTER
A: I had a really good time for taking lectures. B: What classes did you have? A: Well, I had English from **9 o'clock to 11 o'clock**, art from ___, and math from **2 o'clock to 4 o'clock**. B: What do you think about the teachers? A: To be honest, I liked all of them.	12 o'clock to 2 o'clock 11 o'clock to 1 o'clock 12:00 AM to 4:00 AM 7:00 PM to 11:00 PM	✖ ✖ ✔ ✔	✔ ✔ ✖ ✖
A: I have a meeting this afternoon. B: When will it begin? A: It will begin at **three o'clock**. What's the time **now**? B: It is ___. A: I have to go now. I don't want to be late. B: Don't worry, time is enough.	half past one quarter to two half past three half past nine	✖ ✔ ✔ ✖	✔ ✔ ✖ ✖
A: Good morning, What time is it now? B: It is **9 o'clock** now. A: I see. What is today's schedule? B: You have two meetings today. One is at **10 am**, and the other is at ___. After the meeting, you will have dinner with Mr. Brown	2:00 pm 11:00 am 9:00 pm 2:00 am	✖ ✔ ✔ ✖	✖ ✔ ✔ ✖

Table 3: Case study and error analysis of the model predictions. The first two of the four candidate answers are correct.

2019) (what happens next after certain events). As human annotation on the temporal commonsense is costly, many works focus on designing specific continual self-supervised objectives to conduct a temporal commonsense-centric pre-training. For example, (Lin, Chambers, and Durrett 2021) utilizes narrative documents corpus to automatically construct data for temporal ordering and event infilling tasks. (Zhou et al. 2020) jointly models three key dimensions of temporal commonsense (duration, frequency, and typical time) and the other two auxiliary dimensions of TCS, the data of which is mined from unannotated free text. In this work, we focus on reasoning about the temporal expressions over complex context, which also requires understanding temporal commonsense property interwoven with events. However, our work has two notable differences from those works. First, we work on the general temporal expressions of numerical type, which include but are not limited to the time of occurrence, the duration, period, etc, while the works above only focus on the TCS property related to events. Second, we focus on reasoning over *complex* context (e.g., dialog), where the model should be able to disentangle the explicit and implicit inter-dependencies among multiple temporal concepts appearing in the context and reason about the global context. While previous ones usually deal with limited context and focus on a specific temporal concept in isolation.

Logic Rule Induction in NLP

The advantage of logic rule induction is that it combines deep learning's ability on dealing with uncertainty and logic programming's ability for explainable reasoning. Recently, various studies learn logic rules for reasoning on knowledge graphs and multi-hop RC (Qu et al. 2021; Huang et al. 2021; Wang and Pan 2022). Traditional methods enumerate the latent logic rules and further learn a scalar weight to assess the quality of logic rules, while some recent works based on neural logic programming and neural theorem provers can learn logic rules and their weights in an end-to-end manner. We borrow the ideas from RNNlogic (Qu et al. 2021), which treats logic rules as a latent variable and simultaneously trains a rule generator as well as a reasoning predictor with logic rules. However, the rule inducted for temporal reasoning can not be directly grounded, due to the *fuzzy* expressions for temporal concepts in language. We tackle this problem by utilizing the neural network to assess and eliminate the uncertainty of fuzzy temporal concepts.

Conclusion

In this paper, we propose a novel pre-training framework LECTER to augment the language model with temporal logic induction ability for temporal commonsense reasoning. It encourages the model to resolve the underlying relational composition for contextualized temporal commonsense reasoning. The experimental result on the TIMEDIAL dataset demonstrates the efficiency of our proposed pre-training framework, which is better than traditional pre-training methods by a large margin. Our result suggests that the temporal reasoning models can benefit from explicitly modeling the context-dependent temporal logic rules.

Acknowledgments

We would like to thank Li Du for his valuable feedback and advice, and the anonymous reviewers for their constructive comments, and gratefully acknowledge the support of the Technological Innovation "2030 Megaproject" - New Generation Artificial Intelligence of China (2018AAA0101901), and the National Natural Science Foundation of China (62176079, 61976073), and the Industry-University-Research Innovation Foundation of China University (2021ITA05009).

References

Bethard, S.; Derczynski, L.; Savova, G. K.; Pustejovsky, J.; and Verhagen, M. 2015. SemEval-2015 Task 6: Clinical TempEval. In **SEMEVAL*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

Cai, B.; Ding, X.; Chen, B.; Du, L.; and Liu, T. 2022. Mitigating Reporting Bias in Semi-supervised Temporal Commonsense Inference with Probabilistic Soft Logic. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10): 10454–10462.

Chang, A. X.; and Manning, C. D. 2012. SUTime: A library for recognizing and normalizing time expressions. In *LREC*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Elazar, Y.; and Goldberg, Y. 2019. Where's My Head? Definition, Data Set, and Models for Numeric Fused-Head Identification and Resolution. *Transactions of the Association for Computational Linguistics*, 7: 519–535.

Han, R.; Hsu, I.-H.; Sun, J.; Baylon, J.; Ning, Q.; Roth, D.; and Peng, N. 2021. ESTER: A Machine Reading Comprehension Dataset for Reasoning about Event Semantic Relations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 7543–7559. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Han, R.; Ren, X.; and Peng, N. 2021. ECONET: Effective Continual Pretraining of Language Models for Event Temporal Reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 5367–5380. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Han, R.; Zhou, Y.; and Peng, N. 2020. Domain Knowledge Empowered Structured Neural Net for End-to-End Event Temporal Relation Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5717–5729. Online: Association for Computational Linguistics.

Helwe, C.; Clavel, C.; and Suchanek, F. M. 2021. Reasoning with Transformer-based Models: Deep Learning, but Shallow Reasoning. In *AKBC*.

Huang, Y.-X.; Dai, W.-Z.; Cai, L.-W.; Muggleton, S. H.; and Jiang, Y. 2021. Fast Abductive Learning by Similarity-based Consistency Optimization. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 26574–26584. Curran Associates, Inc.

Lange, L.; Iurshina, A.; Adel, H.; and Strötgen, J. 2020. Adversarial Alignment of Multilingual Models for Extracting Temporal Expressions from Text. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, 103–109. Online: Association for Computational Linguistics.

Lee, I.-T.; and Goldwasser, D. 2019. Multi-Relational Script Learning for Discourse Relations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4214–4226. Florence, Italy: Association for Computational Linguistics.

Leeuwenberg, A. M.; and Moens, M.-F. 2018. Temporal Information Extraction by Predicting Relative Time-lines. In *EMNLP*.

Lin, S.-T.; Chambers, N.; and Durrett, G. 2021. Conditional Generation of Temporally-ordered Event Sequences. In *ACL*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv*, abs/1907.11692.

Manhaeve, R.; Dumančić, S.; Kimmig, A.; Demeester, T.; and De Raedt, L. 2021. Neural probabilistic logic programming in DeepProbLog. *Artificial Intelligence*, 298: 103504.

Ning, Q.; Zhou, B.; Wu, H.; Peng, H.; Fan, C.; and Gardner, M. 2022. A Meta-framework for Spatiotemporal Quantity Extraction from Text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2736–2749. Dublin, Ireland: Association for Computational Linguistics.

Qin, L.; Gupta, A.; Upadhyay, S.; He, L.; Choi, Y.; and Faruqui, M. 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)*, 7066–7076. Online: Association for Computational Linguistics.

Qu, M.; Chen, J.; Xhonneux, L.-P.; Bengio, Y.; and Tang, J. 2021. {RNNL}ogic: Learning Logic Rules for Reasoning on Knowledge Graphs. In *International Conference on Learning Representations*.

Raedt, L. D.; Kimmig, A.; and Toivonen, H. T. 2007. ProbLog: A Probabilistic Prolog and its Application in Link Discovery. In *IJCAI*.

Rong, X.; Fourney, A.; Brewer, R. N.; Morris, M. R.; and Bennett, P. N. 2017. Managing Uncertainty in Time Expressions for Virtual Assistants. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, 568–579. New York, NY, USA: ACM. ISBN 978-1-4503-4655-9.

Thukral, S.; Kukreja, K.; and Kavouras, C. 2021. Probing Language Models for Understanding of Temporal Expressions. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 396–406. Punta Cana, Dominican Republic: Association for Computational Linguistics.

Vashishtha, S.; Van Durme, B.; and White, A. S. 2019. Fine-Grained Temporal Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2906–2919. Florence, Italy: Association for Computational Linguistics.

Viani, N.; Tissot, H.; Bernardino, A.; and Velupillai, S. 2019. Annotating Temporal Information in Clinical Notes for Timeline Reconstruction: Towards the Definition of Calendar Expressions. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, 201–210. Florence, Italy: Association for Computational Linguistics.

Wang, W.; and Pan, S. 2022. Deep Inductive Logic Reasoning for Multi-Hop Reading Comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4999–5009. Dublin, Ireland: Association for Computational Linguistics.

Yang, Z.; Du, X.; Rush, A.; and Cardie, C. 2020. Improving Event Duration Prediction via Time-aware Pre-training. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 3370–3378. Online: Association for Computational Linguistics.

Zhou, B.; Ning, Q.; Khashabi, D.; and Roth, D. 2020. Temporal Common Sense Acquisition with Minimal Supervision. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7579–7589. Online: Association for Computational Linguistics.

Zhou, Y.; Geng, X.; Shen, T.; Long, G.; and Jiang, D. 2022a. EventBERT: A Pre-Trained Model for Event Correlation Reasoning. In *Proceedings of the ACM Web Conference 2022*, WWW '22, 850–859. New York, NY, USA: Association for Computing Machinery. ISBN 9781450390965.

Zhou, Y.; Shen, T.; Geng, X.; Long, G.; and Jiang, D. 2022b. ClarET: Pre-training a Correlation-Aware Context-To-Event Transformer for Event-Centric Generation and Classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2559–2575. Dublin, Ireland: Association for Computational Linguistics.