Reading Comprehension with Graph-based Temporal-Casual Reasoning

Yawei Sun, Gong Cheng, Yuzhong Qu
National Key Laboratory for Novel Software Technology, Nanjing University, China
ywsun@smail.nju.edu.cn, gcheng@nju.edu.cn, yzqu@nju.edu.cn

Abstract

Complex questions in reading comprehension tasks require integrating information from multiple sentences. In this work, to answer such questions involving temporal and causal relations, we generate event graphs from text based on dependencies, and rank answers by aligning event graphs. In particular, the alignments are constrained by graph-based reasoning to ensure temporal and causal agreement. Our focused approach self-adaptively complements existing solutions; it is automatically triggered only when applicable. Experiments on RACE and MCTest show that state-of-the-art methods are notably improved by using our approach as an add-on.

1 Introduction

The task of *reading comprehension* has received wide attention from the research community. A machine's ability to understand natural language is tested by answering multiple-choice questions based on a given passage (Hirschman et al., 1999), as illustrated in Fig. 1. State-of-the-art methods have achieved encouraging results on simple factoid questions, but still, it is a challenge to answer a complex question that requires extracting information from more than one sentence in the passage. Typical examples are those *involving temporal or causal relations* spanning multiple sentences (Sugawara et al., 2017; Trischler et al., 2016), like Questions 1 and 2 in Fig. 1. They constitute a considerable proportion of the questions in popular datasets like MCTest (Richardson et al., 2013) and RACE (Lai et al., 2017).

To integrate multiple sentences for answering a complex question, various techniques have emerged, but many of them ignore the semantic relations between sentences (Wang et al., 2016) or superficially use a sliding window scanning over the words of the passage without sentence breaks (Smith et al., 2015; Trischler et al., 2016). To meet the challenge, we *explicitly model the event structure of the passage by a graph representing temporal and causal relations between events* extracted from the passage. Representing a candidate answer in a similar way, we align the two graphs subject to temporal and causal agreement which is verified by *graph-based reasoning*, and rank candidate answers by their degrees of alignment. Our approach, called Graph-based Temporal-Casual Reasoning (GTCR) which focuses on temporal and casual questions, can be used as *an effective add-on to other methods*.

Our research contribution in this paper is threefold.

- To answer complex questions in reading comprehension tasks, we represent the passage and each candidate answer by an event graph to explicitly model temporal and causal relations, which are crucial to the understanding and integration of multiple sentences for answering questions. We generate event graphs from text based on dependencies produced by an off-the-shelf parser, thereby benefiting from methodological progress and tools on dependency parsing.
- To score a candidate answer, we align it with the passage using their event graphs. In an alignment, event matching is constrained by temporal and causal agreement, for which graph-based event reasoning is performed. The alignment problem is formulated and solved using linear programming.

Proceedings of the 27th International Conference on Computational Linguistics, pages 806–817
Santa Fe, New Mexico, USA, August 20-26, 2018.

Passage: Susan held a birthday party. She made a big cake, and hung up some balloons. Meanwhile, her parents bought chocolate ice creams because she enjoyed it. Soon, her friends showed up. Then, Susan hugged her friends. Each friend had a present for Susan. Therefore, Susan was happy and sent each friend a thank you card. So, her friends were happy, too.

Question 1: What did Susan do before her friends came out?
(A) Susan bought ice cream. (B) Susan hung up balloons. (C) Susan hugged her friends. (D) Susan sent friends thank you cards.

Question 2: Why did Susan send out thank you cards to her friends?
(A) Her friends love her. (B) Her friends brought her gifts at the party. (C) Her friends was happy. (D) Her friends came late.

Figure 1: A passage and two questions for reading comprehension.

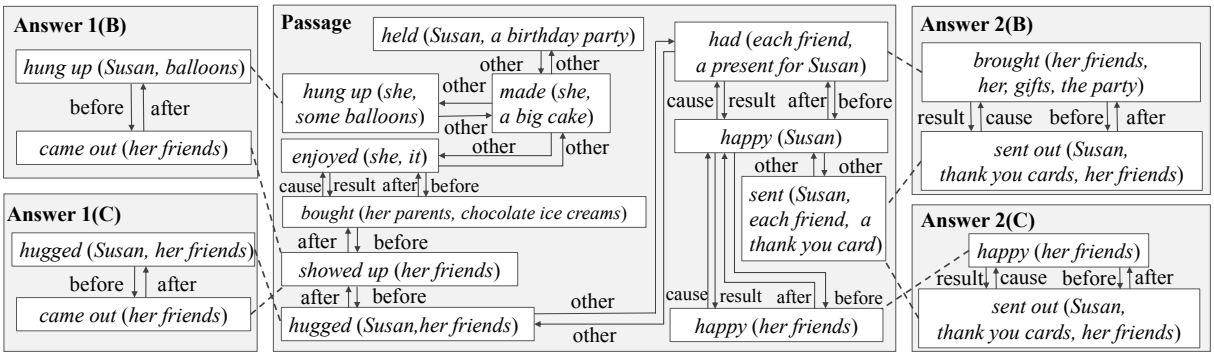

Figure 2: An event graph for the passage and four event graphs for question-answer combinations.

- By automatically detecting question type and determining confidence in scores, our self-adaptive approach is triggered only when appropriate, to enhance existing methods as an add-on. A notable improvement on the state of the art is observed in the experiments on RACE and MCTest.

The paper is structured as follows. Section 2 overviews our approach. Sections 3 and 4 describe event graph generation and alignment, respectively. Section 5 combines our approach with existing methods. Section 6 reports experiments. Section 7 compares related work. Section 8 concludes the paper.

2 Overview of the Approach

Our pipelined GTCR approach is outlined in Fig. 3. The passage is transformed into an event graph called the passage graph. The question and each candidate answer are combined and transformed into an event graph called a QA graph. The two graphs are aligned, and the degree of alignment is assigned as the score of the answer.

2.1 Event Graph Generation

The passage is transformed into an event graph, called the passage graph, representing temporal and causal relations between events extracted from the passage. The transformation firstly recognizes events from each sentence and extracts intra-sentence relations, and then merges the results based on inter-sentence relations. We will define event graph in Section 3.1 and elaborate its generation in Section 3.2.

The question and each candidate answer are combined and transformed in a similar way into an event graph, called a QA graph. We will elaborate this step in Section 3.3.

Example 1 *The passage in Fig. 1 is transformed into the passage graph in Fig. 2. Four out of the eight candidate answers in Fig. 1 are combined with questions and transformed into four QA graphs in Fig. 2.*

2.2 Event Graph Alignment

The passage graph and the QA graph for each candidate answer are aligned. The degree of alignment, which is the maximum sum of the degrees of match between aligned events, is assigned as the score of the answer. Alignment is constrained by the agreement in temporal and causal relations between events, which is verified by graph-based reasoning. We will elaborate this step in Section 4.

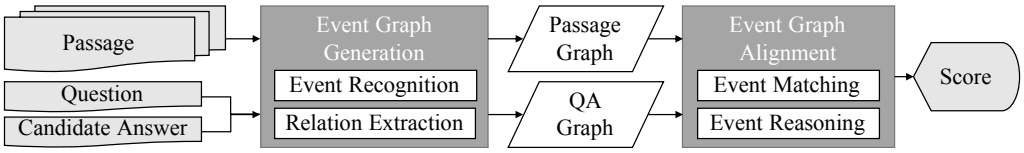

Figure 3: Overview of the GTCR approach.

Example 2 *Dashed lines in Fig. 2 represent exact matches between events, where the alignments for answers 1(C) and 2(C) violate temporal and causal agreement, respectively, thereby being infeasible. The degrees of other (unshown) possible alignments for these answers are not large. Therefore, they are likely to be incorrect answers.*

3 Event Graph Generation

Event graph is a core concept in our approach. In this section we present its definition and detail its generation from the passage. We also describe how to generate an event graph in a similar way from the combination of the question and each candidate answer.

3.1 Event Graph

Definition 1 (Event Graph) *An event graph is a directed multi-graph where vertices represent events which are connected by arcs representing temporal or causal relations between events.*

It is a multi-graph which is permitted to have multiple arcs between two vertices, e.g., both a temporal relation and a causal relation between two events, as illustrated in Fig. 2.

Definition 2 (Event) *An event consists of a predicate and its arguments (Jurafsky and Martin, 2009), and has three attributes (Pustejovsky et al., 2003; Bar-Haim et al., 2015), namely* `aspect` $\in$ $\{$`perfective`, `none`$\}$, `modality` $\in$ $\{$`modal`, `none`$\}$, *and* `polarity` $\in$ $\{$`positive`, `negative`$\}$.

Definition 3 (Relations between Events) *We consider three types of relations.*

Temporal Relation. *An event u can happen* before *another event v, represented by an arc from u to v labeled with* `before`*, and an arc from v to u labeled with* `after`.

Causal Relation. *An event u can* cause *another event v, represented by an arc from u to v labeled with* `result`*, and an arc from v to u labeled with* `cause`.

Other Relation. *We also allow an unspecified relation between two events, represented by a pair of inverted arcs labeled with* `other`.

3.2 Passage Graph Generation

To generate an event graph from the passage, called the *passage graph*, we recognize events from sentences, extract intra-sentence relations to form a local event graph for each sentence, and finally extract inter-sentence relations to merge local event graphs into a global event graph.

3.2.1 Event Recognition

Predicate and Arguments To recognize the predicate and arguments which identify an event, we extend a dependency-based method (Rudinger and Van Durme, 2014). In particular, we consider *embedded events*. Given a dependency relation `ccomp` or `xcomp` whose head is the predicate of an embedding event u and whose dependent is the predicate of an embedded event v, we remove v and assign its predicate and arguments to u's arguments. For example, consider the sentence *Hannah's mother explained that they were moving to Kenya.* The embedded event *moving* is removed; its predicate and arguments, namely *moving* and $\{$*they*, *Kenya*$\}$, become the arguments of the embedding event *explained*.

Attributes To recognize the attributes of an event, we extend a heuristic method (Bar-Haim et al., 2015) which focuses on the attribute `polarity`, and define the following heuristics for recognizing `aspect` and `modality` of an event u based on dependencies.

- Given a dependency relation `aux` whose head is the predicate of u and is POS-tagged with VBN, and whose dependent's lemma is *have*, we have u.`aspect` = `perfective`.
- Given a dependency relation `aux` whose head is the predicate of u and whose dependent is a modal verb (e.g., *will*, *should*), we have u.`modality` = `modal`.

Example 3 *Consider the sentence* Meanwhile, her parents bought chocolate ice creams because she enjoyed it. *As shown in Fig. 2, two events with predicates* bought *and* enjoyed *are recognized. The arguments of* bought *consist of* her parents *and* chocolate ice creams. *The arguments of* enjoyed *consist of* she *and* it. *There is no particular attribute to recognize, so all the three attributes of these events take their default values, i.e.,* `aspect` = `none`, `modality` = `none`, *and* `polarity` = `positive`.

3.2.2 Intra-sentence Relation Extraction

We observe that within a sentence where two or more events are recognized, their relations are usually explicitly indicated by a few syntactic patterns. We define a mapping from dependency patterns to relations as shown in Table 1. Note that some polysemous patterns (e.g., those involving *so*) can give rise to multiple relations. The result is a local event graph generated from each sentence.

Relation	Dependency Pattern
$u \xrightarrow{\texttt{before}} v$ and $u \xleftarrow{\texttt{after}} v$	(i) $u \xrightarrow{\texttt{advcl}} v_{(\texttt{modality=none})} \xrightarrow{\texttt{mark\|advmod}}$ *before|until|till|so|so-that* (ii) $v_{(\texttt{modality=none})} \xrightarrow{\texttt{advcl}} u \xrightarrow{\texttt{mark\|advmod}}$ *after|since|because|for|as* (iii) $v_{(\texttt{modality=none})} \xleftarrow{\texttt{conj}} u \xrightarrow{\texttt{cc}}$ *so*
$u \xrightarrow{\texttt{result}} v$ and $u \xleftarrow{\texttt{cause}} v$	(iv) $u \xrightarrow{\texttt{advcl}} v \xrightarrow{\texttt{mark\|advmod}}$ *so|so-that* (v) $v \xrightarrow{\texttt{advcl}} u \xrightarrow{\texttt{mark\|advmod}}$ *because|for|as* (vi) $v \xleftarrow{\texttt{conj}} u \xrightarrow{\texttt{cc}}$ *so*
$u \xrightarrow{\texttt{other}} v$ and $u \xleftarrow{\texttt{other}} v$	other cases

Table 1: Extraction of intra-sentence relations between two events u and v.

Example 4 *Consider two events* bought *and* enjoyed *recognized in the sentence* Meanwhile, her parents bought chocolate ice creams because she enjoyed it. *Here, two dependency patterns (ii) and (v) in Table 1 are matched, and hence both a temporal relation and a causal relation are extracted, as shown in Fig. 2.*

Example 5 *Consider two events* made *and* hung up *recognized in the sentence* She made a big cake, and hung up some balloons. *An other relation is extracted here, as shown in Fig. 2.*

3.2.3 Inter-sentence Relation Extraction

We extract relations between adjacent sentences, in order to merge local event graphs into a global event graph. For two adjacent sentences, an *exit event* u and an *entrance event* v are selected from the former sentence and the latter sentence, respectively; then relations between u and v are extracted.

Event Selection In the local event graph representing the former sentence, the exit event is one that has `before` or `result` as an incoming arc but never as an outgoing arc. If none of the events in the sentence satisfies this condition, the one that involves the root of the dependency tree (called the *root event*) will be specified as the exit event. Similarly, in the local event graph representing the latter sentence, the entrance event is one that has `before` or `result` as an outgoing arc but never as an incoming arc.

Relation Extraction Having determined the exit event u in the former sentence and the entrance event v in the latter sentence, their relations are extracted in the following way.

- We reuse the set of connective adverbs listed by Lin et al. (2014), and identify two (overlapping) classes: a temporal relation (i.e., $u \xrightarrow{\texttt{before}} v$ and $u \xleftarrow{\texttt{after}} v$) is extracted given *later*, *next*, etc.; or a causal relation (i.e., $u \xrightarrow{\texttt{result}} v$ and $u \xleftarrow{\texttt{cause}} v$) is extracted given *hence*, *therefore*, etc.

- In particular, when $v.\texttt{aspect} = \texttt{perfective}$, a temporal relation in the reverse direction (i.e., $v \xrightarrow{\texttt{before}} u$ and $v \xleftarrow{\texttt{after}} u$) is heuristically extracted.

- In other cases, an other relation (i.e., $u \xrightarrow{\texttt{other}} v$ and $u \xleftarrow{\texttt{other}} v$) is extracted.

Example 6 *Consider two adjacent sentences* Each friend had a present for Susan. Therefore, Susan was happy and sent each friend a thank you card. *The exit event of the former sentence is* had*, and the entrance event of the latter sentence is* happy. *According to the connective adverb* therefore*, both a temporal relation and a causal relation between the two events are extracted, as shown in Fig. 2.*

3.3 QA Graph Generation

We also generate an event graph from the combination of the question q and each candidate answer ans, called a *QA graph*. When q is a fill-in-the-blank question, this QA graph is generated by firstly replacing the blank in q with ans to form a complete text, and then following the method described in Section 3.2 to generate an event graph from this text.

When q is a *wh*-question, we firstly generate an event graph G_q from q using the method described in Section 3.2, and then insert ans into G_q. Here, we differentiate between two types of *wh*-questions.

- For a *why*-question, ans is normally a complete sentence, from which an event graph G_{ans} is generated using the method described in Section 3.2. G_{ans} and G_q are merged by connecting their root events via a causal and a temporal relation, as illustrated by answers 2(B) and 2(C) in Fig. 2.

- For a non-*why*-question, the argument in G_q that contains the interrogative word is replaced by ans. For an exceptional case where ans is a complete sentence and the predicate of the root event in G_q is a general verb like *do* or *happen*, we generate an event graph G_{ans} from ans using the method described in Section 3.2, and then merge G_{ans} and G_q by replacing the event involving the interrogative word in G_q with the root event in G_{ans}, as illustrated by answers 1(B) and 1(C) in Fig. 2.

4 Event Graph Alignment

The degree of alignment between the passage graph G_p and the QA graph G_{qa} for each candidate answer ans is assigned as the score of ans. In this section we formulate and solve the alignment problem, which maximizes event matching and is constrained by graph-based event reasoning.

4.1 Problem Formulation

We define the *degree of alignment* between G_p and G_{qa} as the maximum sum of the degrees of match between aligned events in G_p and G_{qa}, which is formulated as a variant of the maximum weighted bipartite matching problem, which in turn is a special case of a 0-1 linear program.

Specifically, let V_p and V_{qa} be the events in G_p and G_{qa}, respectively. For $u \in V_p$ and $v \in V_{qa}$, let $\texttt{dom}(u, v)$ be the degree of match between u and v, which will be detailed in Section 4.2. The score of ans, i.e., the degree of alignment between G_p and G_{qa}, is given by

$$\begin{aligned} &\text{maximizing} \sum_{u \in V_p, v \in V_{qa}} \texttt{dom}(u, v) x_{uv} \\ &\text{subject to} \sum_{v \in V_{qa}} x_{uv} \leq 1 \text{ for } u \in V_p\,, \quad \sum_{u \in V_p} x_{uv} \leq 1 \text{ for } v \in V_{qa}\,, \\ &\qquad x_{uv} + x_{km} \leq 1 \text{ for } u, k \in V_p,\ v, m \in V_{qa},\ \texttt{conflict}(u, v, k, m) = 1\,, \\ &\qquad x_{uv} \in \{0, 1\} \text{ for } u \in V_p,\ v \in V_{qa}\,, \end{aligned} \tag{1}$$

where $x_{u,v}$ is a binary variable representing the alignment between u and v, taking value 1 if they are aligned, and $\texttt{conflict}(u, v, k, m)$ indicates whether two matches u-v and k-m lead to a conflict in terms of temporal or causal relations between events, which will be detailed in Section 4.3. We further normalize the score of ans by dividing it by the number of events in G_{qa}.

Note that as a special case, we will define the degree of alignment as one minus the above result if the `polarity` of the root event in the question is `negative`.

4.2 Event Matching

The degree of match between two events u and v, denoted by $\texttt{dom}(u, v)$, measures to what extent u and v refer to the same event. For example, each dashed line in Fig. 2 represents an exact match between two coreferential events. Our measure is a combination of element-level matching and word-level matching:

$$\texttt{dom}(u,v) = \begin{cases} \alpha \cdot \texttt{dom}_{\text{elem}}(u,v) + \beta \cdot \texttt{dom}_{\text{word}}(u,v) & \text{if } u.\texttt{polarity} = v.\texttt{polarity}, \\ 0 & \text{if } u.\texttt{polarity} \neq v.\texttt{polarity}, \end{cases} \tag{2}$$

where α, β are weights to be empirically tuned. Note that for two events having opposite values of `polarity`, their degree of match is simply fixed to 0.

Element-level Matching Let $\texttt{Elem}(u)$ be the set of *elements* of event u, consisting of the predicate and arguments of u. For two elements $e_i \in \texttt{Elem}(u)$ and $e_j \in \texttt{Elem}(v)$, we extend the method used by Li and Srikumar (2016) and define 16 features that calculate the similarity between e_i and e_j. Each feature ϕ returns a similarity value in the range of $[0, 1]$. Inspired by Yih et al. (2013), we calculate the overall similarity between e_i and e_j as follows:

$$\texttt{sim}(e_i, e_j) = \max_{\phi} w_\phi \cdot \phi(e_i, e_j)\,, \tag{3}$$

where ϕ is weighted by w_ϕ. The following features and weights are used.

- $w = 1.0$: exact string match, exact string match between roots of dependency trees, existence of trigram overlap, coreference, apposition, sharing synonyms in Wiktionary, sharing synonyms in WordNet, sharing derivations in WordNet, VerbOcean-based similarity;
- $w = 0.5$: Jaccard similarity of words, partial string match, existence of bigram overlap, cosine similarity of Word2vec embeddings, cosine similarity of GloVe embeddings, light-verb-to-verb match;
- $w = 0.1$: existence of unigram overlap.

With `sim` values for all e_i-e_j pairs in $\texttt{Elem}(u) \times \texttt{Elem}(v)$, we find a maximum weighted bipartite matching between $\texttt{Elem}(u)$ and $\texttt{Elem}(v)$, and assign the squared mean weight of this matching to $\texttt{dom}_{\text{elem}}(u, v)$.

Word-level Matching Complementary to element-level matching which relies on dependency-based argument recognition, we measure the degree of word-level match between u and v using a bag-of-words method (Sultan et al., 2016), and assign the result to $\texttt{dom}_{\text{word}}(u, v)$.

4.3 Event Reasoning

For events $u, k \in V_p$ and $v, m \in V_{qa}$, $\texttt{conflict}(u, v, k, m) = 1$ means if two matches u-v and k-m appear in the same alignment, a *conflict* in terms of temporal or causal relations will emerge; such an alignment is not allowed. We consider two types of conflicts by extending the idea in Berant et al. (2014).

Temporal Conflict There is a (directed simple) u-k path in G_p and a v-m path in G_{qa} such that `before` appears in one of them and `after` appears in the other.

Causal Conflict There is a u-k path in G_p and a v-m path in G_{qa} such that `result` appears in one of them and `cause` appears in the other.

For efficiency reasons, our implementation only checks paths that contain not more than 10 arcs.

Example 7 *The alignment represented by the dashed lines between the passage graph and the QA graph for answer 1(C) in Fig. 2 is not feasible due to the temporal conflict induced by the two matches. The alignment for answer 1(B) is feasible. Similarly, the alignment for answer 2(C) is not feasible due to a causal conflict, whereas the alignment for answer 2(B) is feasible.*

5 Combination with Other Methods

Our GTCR approach is specifically developed for temporal and causal questions. Reading comprehension tasks also involve other questions. Therefore, GTCR is to be combined with other methods.

We propose to let GTCR *self-adaptively enhance existing methods as an add-on*. Specifically, GTCR will answer a question only if it accepts the type of the question and is confident in the scores it generates. Otherwise, it will not answer the question, which in turn will be answered by an existing method. In this section we describe how to *automatically* detect question type and determine confidence in scores.

5.1 Question Type

Our current implementation of GTCR can process *wh*-questions and fill-in-the-blank questions. However, it only accepts temporal and causal questions. A temporal question, e.g., Question 1 in Fig. 1, is one that matches dependency patterns (i) or (ii) in Table 1, or in case there are errors in dependency parsing, matches the following POS-level regular expression: .*VB.+(*when*|*after*|*before*|*since*).+VB.*. A causal question, e.g., Question 2 in Fig. 1, is one that begins with *why*, or contains an adverbial clause fronted by *because* (which is common for fill-in-the-blank questions).

5.2 Confidence in Scores

As GTCR relies on the results of dependency parsing, it will not be confident in the scores it generates if it finds errors in the dependency tree of the question. Specifically, inspired by Petrov et al. (2010), our implementation detects five common types of errors in dependency parsing:

- `nsubj`: missing a subject in a question or clause,
- `aux`: placing an auxiliary verb as the root of the dependency tree,
- `dep`: associating `dep` with a subject,
- `acl:relcl`: accompanying `acl:relcl` which indicates a relative clause with a conjunction commonly fronting an adverbial clause (e.g., *when*), and
- `advcl`: missing core arguments (e.g., `nsubj`, `obj`) in an adverbial clause.

When dependency parsing seems correct, GTCR will still reject the question if the score of the answer it selects is below a threshold τ, which will be tuned in the experiments.

6 Experiments

6.1 Datasets

Experiments are performed on two standard datasets for reading comprehension: RACE and MCTest.

RACE (Lai et al., 2017) consists of nearly 28,000 passages and 100,000 questions collected from English exams for middle and high school Chinese students. In particular, 25.8% of these questions require multi-sentence reasoning, though not necessarily temporal or causal reasoning. For our experiments we use middle school data, denoted by RACE-M.

MCTest (Richardson et al., 2013) consists of 660 fictional stories as passages and 2,640 questions that a seven-year-old can understand. It is divided into MC160 where 160 stories have been manually curated for quality, and MC500 where 500 stories may have grammatical or other errors.

6.2 Competing/Collaborating Methods

For RACE we consider two state-of-the-art methods. Stanford Attentive-Reader (Stanford AR) (Chen et al., 2016) is a neural network system based on the Attentive-Reader model. Gated-Attention (GA) Reader (Dhingra et al., 2017) integrates a multi-hop architecture with an attention mechanism.

For MCTest we consider three state-of-the-art methods. Smith et al. (2015) propose a lexical matching method that takes into account multiple context windows, question types, and coreference resolution.

Narasimhan and Barzilay (2015) incorporate discourse information. Trischler et al. (2016) harness simple neural networks arranged in a parallel hierarchy.

Some other methods are not considered in our experiments because their codes or detailed results are not available at the time of experiment so that we could not test the combination of GTCR with them.

6.3 Our Approach

For our implementation of GTCR, we use Stanford CoreNLP (Manning et al., 2014) for dependency parsing and coreference resolution, and use Gurobi (www.gurobi.com) for solving linear programs.

Some parameters are tuned based on the development set in each dataset, which is separate from the test set. Specifically, for Eq. (2) we set $\alpha = \beta = 0.5$ for all the datasets. The confidence threshold τ used for combination with other methods is set to 0.00 for RACE, 0.71 for MC160, and 0.78 for MC500.

Our implementation is open source[1].

	RACE-M
Stanford AR	58/133
GA Reader	50/133
GTCR	**63/133**
GTCR w/o elem. matching	63/133
GTCR w/o word matching	59/133
GTCR w/o reasoning	61/133

Table 2: Accuracy of competition on RACE-M.

	MC160	**MC500**
Smith et al. (2015)	8/13	9/16
Narasimhan et al. (2015)	7/13	8/16
Trischler et al. (2016)	—	10/16
GTCR	**9/13**	**12/16**
GTCR w/o elem. matching	7/13	9/16
GTCR w/o word matching	10/13	11/16
GTCR w/o reasoning	9/13	10/16

Table 3: Accuracy of competition on MCTest.

	RACE-M	
		+GTCR
Stanford AR	44.15	**44.50**
GA Reader	43.73	**44.64**

Table 4: Accuracy (%) of collaboration on RACE-M.

	MC160		**MC500**	
		+GTCR		+GTCR
Smith et al. (2015)	74.27	**74.68**	65.79	**66.29**
Narasimhan et al. (2015)	71.45	**72.29**	63.71	**64.38**
Trischler et al. (2016)	74.58	—	70.41	**70.75**

Table 5: Accuracy (%) of collaboration on MCTest.

6.4 Results

We carry out two experiments. In the first experiment, GTCR *competes* with state-of-the-art methods on complex questions that require temporal or causal reasoning. In the second experiment, GTCR self-adaptively *collaborates* with those methods as an add-on, in order to advance the state of the art.

We measure the accuracy of each method, namely the proportion of correctly answered questions.

Competition On RACE-M, among the 1,436 test questions, 133 are recognized as temporal or causal questions according to question type detection in GTCR. Table 2 shows the accuracy of each method on these questions. Compared with competing methods, GTCR correctly answers 4–10% more of those questions.

On MCTest, among the 240 and 600 test questions in MC160 and MC500, 51 and 99 respectively are recognized as temporal or causal questions. After excluding 10 and 17 questions respectively having dependency errors and those on which GTCR is not confident, Table 3 shows the results on the remaining questions. GTCR notably outperforms competing methods on these questions.

Ablation To evaluate the effectiveness of our event matching and reasoning, we conduct an ablation study. As shown in Tables 2 and 3, the accuracy of our approach decreases by up to 19% and 6% when element-level and word-level event matching are disabled, respectively, showing that the two techniques

[1] http://ws.nju.edu.cn/qa/coling2018

are generally useful; only for one question in MC160, word-level matching has a negative effect. Without performing event reasoning, the accuracy decreases by 2% on RACE-M and by 13% on MC500, showing its usefulness. However, reasoning is not helpful on MC160.

Collaboration By self-adaptively collaborating with existing methods as an add-on, GTCR consistently enhances all the methods and improves the state of the art on both datasets. As shown in Table 4, the overall accuracy is increased by 0.35–0.91% on RACE-M. For MCTest in Table 5, the improvement is in the range of 0.41–0.84% on MC160, and 0.34–0.67% on MC500. On MC160 we do not add GTCR to Trischler et al. (2016) because at the time of experiment we did not obtain its details results.

6.5 Error Analysis and Discussion

In GTCR, errors may occur in event graph generation and event matching. We analyze the 78 questions in Tables 2 and 3 for which GTCR produces incorrect answers.

Event Graph Generation Although GTCR can automatically detect a few common types of errors in dependency parsing, other POS or dependency errors are still observed in the experiments, which lead to imprecise recognition of events and extraction of relations (18%). These errors are propagated along the pipeline framework of GTCR, which has the potential to be significantly improved by benefiting from future advances in dependency parsing.

Another drawback to GTCR is the use of syntactical methods for extracting explicit relations between events. Some temporal and causal relations are implicit, and hence are missing in our generated event graphs (31%). To identify those relations, commonsense knowledge is to be exploited.

Event Matching Event matching is closely related to paraphrasing. Our element-level and word-level matching in GTCR, though utilizing many linguistic resources, still fail to identify some coreferential events that are expressed differently (19%). It inspires us to integrate or develop more effective techniques for paraphrasing and textual entailment in future work.

Another disadvantage of GTCR is its limited expressivity of event alignment. We assume one-to-one event matching between event graphs, but in practice it is sometimes one-to-many (6%).

6.6 Running Time

On an Intel E3-1225v3 (3.2GHz), our approach uses 7.8 seconds, 6.5 seconds, and 7.0 seconds to process a question in RACE-M, MC160, and MC500, respectively, being reasonably fast. Most time is used for: retrieving external resources in element-level event matching, and identifying conflicting event matches in event graph alignment.

7 Related Work

7.1 Reading Comprehension

Many solutions to reading comprehension are learning-based. They either manually define and extract features for learning a model to predict the correct answer (Sachan et al., 2015; Smith et al., 2015; Wang et al., 2015; Sachan and Xing, 2016), or use deep learning with attention mechanisms (Hermann et al., 2015; Trischler et al., 2016; Wang et al., 2016; Chen et al., 2016; Chen et al., 2017; Cui et al., 2017; Dhingra et al., 2017; Samothrakis et al., 2017). These methods have achieved encouraging results on simple factoid questions, but are not optimized for complex questions that require integrating information from multiple sentences. By comparison, our approach is distinguished by the explicit representation and reasoning over temporal and causal relations between events which can span multiple sentences, therefore exactly targeting complex questions and featuring an explainable problem solving process. Due to its unsupervised nature, our approach can be rapidly deployed for a new domain or application.

We are not the first to address complex questions. Sachan and Xing (2016) leverage Abstract Meaning Representation (AMR) to construct graph representations and solve a graph containment problem, but they report that their method suffers from the lack of complex reasoning support (e.g., temporal, causal), which are provided in our approach. Narasimhan and Barzilay (2015) incorporate discourse information

into a discriminative framework with hidden variables that capture relevant sentences and their relations, which is different from our explicit representation of events and their relations extracted from text, and is inferior to our approach in the experiments. Berant et al. (2014) model the relations between events in a biological process using a graph representation, which is similar to our approach. However, to generate a graph, they train dedicated classifiers whereas we use dependencies produced by off-the-shelf parsers. Besides, for question answering, they simply use a few hand-crafted regular expressions over graphs whereas we more thoughtfully formulate and solve a maximum weighted bipartite matching problem.

7.2 Event Extraction

Event extraction has attracted considerable research attention (Pustejovsky et al., 2003; Ahn, 2006; Rudinger and Van Durme, 2014; Mostafazadeh et al., 2016; Mirza and Tonelli, 2016; Judea and Strube, 2016). We extend some of those existing methods (Rudinger and Van Durme, 2014) for event recognition. As to relation extraction, our rule-based method uses dependencies, whereas there are also learning-based solutions (Ahn, 2006; Chambers et al., 2007; Mirza and Tonelli, 2016). Although our focus is on the reading comprehension task, event extraction is at the core of our approach. Advances in this research area will be beneficial to the performance of our approach.

8 Conclusion

To answer complex questions in reading comprehension tasks, we generate event graphs from text for modeling temporal and causal relations spanning multiple sentences. With graph-based reasoning, our self-adaptive GTCR approach helps advance the state of the art on two standard datasets, showing the effectiveness and generalizability of the proposed technical contribution. Furthermore, this graph-based event representation and reasoning complies with human cognition, thereby being capable of not only answering a question but also providing a human-readable explanation. This will expand the scope of application, ranging from explainable question answering to interactive computer-aided education.

Many components of GTCR build upon the results of dependency parsing. Featuring an unsupervised framework, our proposed approach can be conveniently transferred to different domains and applications without the necessity of preparing dedicated training data. Besides, its performance will progress along with the improvement in off-the-shelf dependency parsers. However, as revealed by our post-experiment error analysis, there are still several other directions for improving and extending our approach, to turn it into a more effective add-on to other methods or even a powerful stand-alone solution.

Acknowledgments

This work was supported in part by the NSFC under Grants 61772264 and 61572247, and in part by the Qing Lan and Six Talent Peaks Programs of Jiangsu Province.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8. Association for Computational Linguistics.

Roy Bar-Haim, Ido Dagan, and Jonathan Berant. 2015. Knowledge-based textual inference via parse-tree transformations. *J. Artif. Intell. Res.(JAIR)*, 54:1–57.

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D Manning. 2014. Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1499–1510.

Nathanael Chambers, Shan Wang, and Dan Jurafsky. 2007. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 173–176. Association for Computational Linguistics.

Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2358–2367.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1870–1879.

Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2017. Attention-over-attention neural networks for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 593–602.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2017. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1832–1846.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Lynette Hirschman, Marc Light, Eric Breck, and John D Burger. 1999. Deep read: A reading comprehension system. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 325–332. Association for Computational Linguistics.

Alex Judea and Michael Strube. 2016. Incremental global event extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2279–2289.

Dan Jurafsky and James H Martin. 2009. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Tao Li and Vivek Srikumar. 2016. Exploiting sentence similarities for better alignments. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2193–2203.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A pdtb-styled end-to-end discourse parser. *Natural Language Engineering*, 20(2):151–184.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Paramita Mirza and Sara Tonelli. 2016. Catena: Causal and temporal relation extraction from natural language texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 64–75.

Nasrin Mostafazadeh, Alyson Grealish, Nathanael Chambers, James Allen, and Lucy Vanderwende. 2016. Caters: Causal and temporal relation scheme for semantic annotation of event structures. In *Proceedings of the Fourth Workshop on Events*, pages 51–61.

Karthik Narasimhan and Regina Barzilay. 2015. Machine comprehension with discourse relations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1253–1262.

Slav Petrov, Pi-Chuan Chang, Michael Ringgaard, and Hiyan Alshawi. 2010. Uptraining for accurate deterministic question parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 705–713. Association for Computational Linguistics.

James Pustejovsky, José M Castano, Robert Ingria, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. 2003. Timeml: Robust specification of event and temporal expressions in text.

Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 193–203.

Rachel Rudinger and Benjamin Van Durme. 2014. Is the stanford dependency representation semantic? In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 54–58.

Mrinmaya Sachan and Eric Xing. 2016. Machine comprehension using rich semantic representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 486–492.

Mrinmaya Sachan, Kumar Dubey, Eric Xing, and Matthew Richardson. 2015. Learning answer-entailing structures for machine comprehension. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 239–249.

Spyridon Samothrakis, Tom Vodopivec, Michael Fairbank, and Maria Fasli. 2017. Convolutional-match networks for question answering. International Joint Conferences on Artificial Intelligence.

Ellery Smith, Nicola Greco, Matko Bosnjak, and Andreas Vlachos. 2015. A strong lexical matching method for the machine comprehension test. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1693–1698. Association for Computational Linguistics.

Saku Sugawara, Hikaru Yokono, and Akiko Aizawa. 2017. Prerequisite skills for reading comprehension: Multi-perspective analysis of mctest datasets and systems. In *AAAI*, pages 3089–3096.

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2016. Dls @ cu at semeval-2016 task 1: Supervised models of sentence similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 650–655.

Adam Trischler, Zheng Ye, Xingdi Yuan, Jing He, and Philip Bachman. 2016. A parallel-hierarchical model for machine comprehension on sparse data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 432–441.

Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2015. Machine comprehension with syntax, frames, and semantics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 700–706.

Bingning Wang, Shangmin Guo, Kang Liu, Shizhu He, and Jun Zhao. 2016. Employing external rich knowledge for machine comprehension. In *IJCAI*, pages 2929–2925.

Wen-tau Yih, Ming-Wei Chang, Christopher Meek, and Andrzej Pastusiak. 2013. Question answering using enhanced lexical semantic models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1744–1753.