Evaluating Temporal Information Understanding with Temporal Question Answering

Naushad UzZaman♣
University of Rochester
Rochester, New York, USA
naushad@cs.rochester.edu

Hector Llorens♣
University of Alicante
Alicante, Spain
hllorens@dlsi.ua.es

James Allen
University of Rochester
Rochester, New York, USA
james@cs.rochester.edu

***Abstract*—The temporal annotation scheme TimeML was developed to support research in complex temporal question answering (QA). Given the complexity of temporal QA, most of the efforts have focused, so far, on extracting temporal information, which has been evaluated with corpus-based evaluation. However, the QA task represents a natural way to evaluate temporal information understanding, and creating question sets is less costly for humans than manually annotating temporal information, which is required to perform corpus-based evaluation. Additionally, QA performance better captures the understanding of important temporal information as compared to corpus-based evaluation where all information is equally important for scoring. This paper presents a temporal QA system that performs temporal reasoning. It can be used to answer temporal questions (*factoid, list* and *yes/no*), about any document annotated in TimeML. In the paper, we show how this system can be used to evaluate automated temporal information understanding. Our QA-based evaluation results suggest that (i) the available temporal annotations are not complete, and (ii) QA provides a less costly and more reliable way of evaluating temporal understanding systems. To favour replicability, we made the temporal QA system and the question set used in the evaluation available.**

I. Introduction and Motivation

TimeML [1] is a scheme to annotate temporal information (temporal expressions, events, and their relations) in natural language texts. The motivation of TimeML was to serve as basis to tackle complex temporal question answering (QA). Given the complexity of temporal QA, the focus was shifted to solve the automated TimeML annotation task itself. This task implies smaller subtasks such as extracting events, temporal expressions or identifying temporal relations between them. Currently, automated systems solving these subtasks are evaluated using corpus-based evaluation [2].

Corpora annotated in TimeML are certainly needed for developing and training automated systems. However, we argue that measuring how well automated systems understand the temporal aspects of language can be better done through QA. Our main arguments in favor of this task-based evaluation are the following:

1) Answering questions is a natural way of evaluating language understanding for humans. Since humans ask questions about relevant information, QA scores better capture the understanding level of important temporal information, as compared to corpus-based evaluation where all information is equally important for scoring.
2) Creating temporal questions is much easier and less time-consuming for humans than annotating temporal information. Furthermore, providing the correct answers for such questions is much more reliable than providing a correct and a complete temporal annotation as it is explained later in the paper. QA also makes it possible to easily create large test-sets and to evaluate the generality of systems in new domains.
3) Human annotations can be incomplete. Evaluating automated systems against incomplete annotation might not reflect the actual performance of automated systems. With QA, we can also evaluate: (i) how complete the available human annotations are and (ii) how automated systems are performing compared to human annotations.

In this paper, we analyze temporal QA performance as temporal information understanding evaluation measure. For that purpose, we present a temporal QA system that handles temporal reasoning and allows answering complex temporal questions about any text annotated with TimeML.

II. Related Work

Following Allen [3], in this paper, we assume that temporal entities (events and temporal expressions) are represented by time-intervals, and each pair of entities can present one of the following thirteen relations: *before, after, overlap, overlappedBy, start, startedBy, finish, finishedBy, during, contains, meet, metBy* and *simultaneous*. We take the following definition of temporal reasoning: "Given a set of explicit temporal relations between a set of entities, temporal reasoning infers additional relations between entities that are implicit in the ones given".

There are related work on temporal QA systems that perform temporal reasoning. Below we briefly describe these systems and their limitations. After that we include a short review about the temporal evaluation.

1) Temporal QA Systems: Temporal QA implies answering questions such as listing which events happened after some event or time, inferring when some event occurred in time or with respect to other events, etc.

Pustejovsky et al. [4] and Hobbs and Pustejovsky [5] discuss how TimeML could be used for temporal QA, but none of

♣The first and the second author contributed equally to this paper.

them present a system to solve the task.

Harabagiu and Bejan's work [6] on QA with temporal inference handles question-answering based on general inference but not temporal reasoning in pure sense, as defined previously. The following example briefly explains their approach. Given the question Q1: "When did Iraq invade Kuwait?" The answer to Q1 is "2 August 1990", extracted from the context: "Iraqis have been struggling under UN sanctions ever since Hussein's annexation of Kuwait on 2 August 1990." Harabagiu and Bejan identify temporal relations between entities with temporal signals and then make semantic inference to match the question with the answer context, e.g. matching Iraq's invasion with Hussein's annexation.

Similarly, Moldovan et al.'s work [7] on temporal reasoning approach the problem by identifying the temporal relations between entities and feed a module in their overall inference system. They do inference between temporal expressions but not events. Both of these systems ([6] and [7]) are therefore unable to do complete temporal reasoning and answer temporal questions about events not directly related to times.

Chambers and Jurafsky [8] use transitive closure properties as global inference to identify temporal relations between two entities. In the process, they can identify how two entities are temporally related, if they are connected. However, their reasoning is limited to *before, after* and *vague* relations, whereas, there are 13 temporal relations [3].

The web-based question-answering systems are not discussed, since they rely heavily on answer redundancy instead of temporal reasoning [9], [10].

2) Evaluation of temporal information understanding: The main-stream QA forum TREC[1] evaluated NLP systems mainly on *factoid questions*, *list questions* and *other questions*.

In terms of temporal evaluation, influenced by temporal evaluation shared tasks [2], only the corpus-based evaluation has been explored. Nevertheless, task based evaluations have not been proposed for temporal information processing before.

In this paper we explore QA to evaluate temporal information understanding. To achieve our goal, we present an end-to-end advanced temporal QA system performing temporal reasoning. Unlike related works, we handle all the 13 temporal relations, instead of a subset. Like TREC, we consider *factoid*, and *list questions*. However, given our focus on temporal reasoning, we also include *yes-no question* category.

III. Temporal QA with Temporal Reasoning

A. Question Type

We implemented a temporal QA system to address the following question types.

1) a) *yes/no*: "Was Fein called after the killings?"
   b) *list*: "What happened after the crash?" "What happened between the crash and yesterday?"
   c) *when (factoid)*: "When did DT Inc. holders adopt a shareholder-rights plan?"

[1]http://trec.nist.gov/

To answer these questions the system has to be aware of the temporal relations between events and temporal expressions, which can be explicit in the TimeML annotation or implicit (inferred by temporal reasoning).

B. TimeML

Given a text in natural language, understanding the temporal information requires anchoring and ordering the events of the text in time. This task involves the extraction of temporal expressions (e.g., 1999, last year, 5 hours), events (e.g., said, war, etc.), and their temporal relations (war time-span is included in 1999 time-span). TimeML [1] defines how to annotate these elements. In particular, TimeML temporal relations are derived from Allen interval relations.

C. Temporal QA with Timegraph

Given a text annotated with TimeML, temporal reasoning can be performed to infer the implicit temporal relations. For example for answering question (1a) above, if it is annotated that *calling Fein* took place after the *killings* then we have an explicit relation pointing out the answer. However, if it is annotated that *Fein took part in the killings*, then *he was captured*, and finally *he was called*. The fact that *he was called* after the *killings* must be inferred through its temporal relation with being *captured*.

Temporal reasoning requires computing all the temporal relations between all the entities of each document. This is a computationally complex and expensive task [3]. From the available algorithms to carry out temporal reasoning Timegraph [11] is one of the most efficient.

We implemented a system to generate a timegraph from a TimeML annotation. Our implementation is an extension of timegraph implementation by UzZaman and Allen [12].

The timegraph is created by adding the TimeML explicit relations. With the timegraph's reasoning mechanism, the derived implicit relations are inferred. Timegraph allows therefore answering questions about both explicit and implicit temporal relations. It can answer questions about any of the Allen relations. To answer the questions about TimeML entities (based on time intervals) using timegraph (based on time points), we convert the queries to point based queries.

For answering yes/no questions, we check the necessary point relations in timegraph to verify an interval relation. For example, to answer the question *is event1 after event2*, our system verifies whether *start(event1)* $>$ *end(event2)*; if it is verified then the answer is *true*, if it conflicts with the timegraph then it is *false*, otherwise it is *unknown*.

For answering list and factoid questions, the system traverses the timegraph. For example, if we want to list all events before *event1*, our query to timegraph would be to find all events that end before the start of *event1*.

The following example illustrates the different steps. Assume, we have some temporal relations (shown in Fig. 1) from a document. Events (e) and temporal expressions (t) are identified by numbers.

e1 AFTER e2	e4 AFTER e6
e1 AFTER e3	e6 SIMULTANEOUS e7
e1 SIMULTANEOUS e4	e6 AFTER t1
e4 SIMULTANEOUS e5	e9 IS-INCLUDED t1

Fig. 1. Some temporal relations from a document

Timegraph stores temporal ordering of entities representing them as time-point pairs: start (s), end (e) with numeric values in different chains (e.g., *event 1 start* $\rightarrow$ *e1s*). The timegraph for the temporal relations in Fig. 1 is shown in Fig. 2.

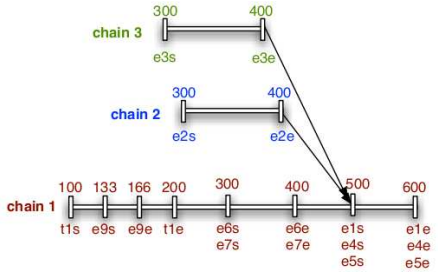

Fig. 2. Timegraph representation of relations in Figure 1

Timegraph permits answering how two entities are related with each other. As a result, *yes/no*, *list* and *factoid* questions can be answered.

Currently, our system uses a specific syntax that represents human language questions. We do not convert natural language questions to our question syntax automatically, instead we input the questions in timegraph query syntax. Below we show examples of the syntax for each type of questions.

For answering the *yes/no* question *Was Fein called after the killings?* the timegraph query will be `IS e1 AFTER e9`. We can see in the relations in Fig. 1 that this particular relation is not explicitly mentioned, but with temporal reasoning we will be able to answer this implicit relation. In our timegraph (Fig. 2), both e1 and e9 are in the same chain 1. Hence we will just check if e1s $>$ e9e, which is true, to answer this question.

We can also answer *list* questions by traversing the timegraph. If we ask *What happened before calling Sinn Fein?* the timegraph query will be `LIST BEFORE e1`. We will traverse through the timegraph, in all chains to answer $\{\forall$ *ei*: eie $<$ e1s$\}$. In this case: e2, e3, e6, e7, t1 and e9.

For *factoid* question, we answer all entities that are simultaneous or are included in the entity. If we ask *When did the killing happened?* the timegraph query will be `WHEN e9`. Our system will be able to answer during t1 (last week) by traversing the timegraph. In this case, we will check all entities (temporal expression and events) *ei*, such that, $\{\forall$ *ei*: eis $\leq$ e9s and eie $\geq$ e9e$\}$.

It is important to note that, if the correct and complete TimeML annotation is provided, this system will retrieve the correct answer for any question in the described syntax.

In order to analyze temporal QA as temporal information understanding evaluation, we carry out a set of experiments.

IV. Temporal QA as Evaluation

The aim of the evaluation is comparing, for temporal information, classical corpus-based vs. QA-based evaluation.

In corpus based evaluation, we compare the performance of gold annotations and automated systems, of which we consider TIPSem [13] and TRIOS [14] from TempEval-2 [2]. The measure used is UzZaman and Allen's [12] metric, which gives a single score for corpus-based evaluation (F1).

In QA-based evaluation, we can compare the answers given by humans, those extracted from gold and automated annotations. The measure used is accuracy (an answer can only be correct or incorrect). Two volunteers randomly selected 25 TimeML annotated documents from TimeBank and, by reading just the text i.e., without looking at the TimeML annotation, they created a set of 189 temporal questions (79 *yes/no*, 63 *list* and 47 *factoid questions*). Questions capture important time-related information from the documents rather than all the temporal information as in corpus-based case.

A. Using gold annotation as reference

We evaluate with QA how accurate the annotations of automated systems are using gold annotation as reference to obtain answers. Then, we compare these answers with those obtained from automated annotations. Table I reports the QA results in addition to the corpus-based results.

	Yes/No	List	Factoid	Corpus-based F1 score
TIPSem	48.10%	43.49%	53.30%	31.60%
TRIOS	34.18%	37.03%	22.04%	27.16%

TABLE I
Temporal QA and corpus-based score against gold annotation

We find that TIPSem does better than TRIOS in every category of questions which matches corpus-based score. However, we can see that TIPSem does significantly better than TRIOS in factoid questions. Corpus-score cannot distinguish how well systems perform specific real tasks like answering factoid questions. With temporal QA, we can evaluate the detailed capabilities of systems.

We also find that reasoning is required to perform QA over TimeML. To determine this, we check if we can answer a yes/no question directly from the annotated relations (explicit relations) or whether we have to use timegraph (implicit relations). Out of our 79 yes/no questions, the gold annotation was capable of answering only 42 questions in the first place. Out of these 42 questions, we found only 7 (16.67%) were explicitly annotated and rest 35 out of 42 (83.3%) were answered with temporal inference that just semantic inference can not answer. The rest were *unknown*, i.e. the human annotation did not provide the information needed to make the inference.

B. Gold annotation vs. human answers

In this experiment we manually answered the yes/no questions and compared the answers with those obtained from systems and gold annotations to understand the coverage of automated and gold temporal annotations. The results obtained are reported in Table II.

We found that performance for gold annotation is not 100% (it is only 48%). This is because the gold annotation does not have a complete temporal information coverage, i.e. it

	Gold annot.	TIPSem	TRIOS
Comparing Yes/No answers against human answers	48.10%	37.97%	22.78%
Comparing Yes/No answers against TimeML gold annotations	100%	48.10%	34.17%

TABLE II
GOLD AND SYSTEM ANNOTATIONS AGAINST HUMAN ANSWERS

does not have all necessary relations to make the reasoning required to answer some of the questions. Note that with our QA system, an annotation with complete coverage will have a 100% performance for these questions.

When comparing against human answers, the difference between the gold and system performance is much lower. This result is suggestive that automated systems can perform very close to human annotations currently available.

We also notice that all scores decrease from the performance reported in Table I. This is because many *yes/no questions* cannot be answered from the gold annotation (answer is *unknown*) – the gold annotation did not have all necessary relations to make the inference. When the systems also answered *unknown* in those cases, they had a match. Hence, when compared against the exact human answers (usually not *unknown*), all the scores decreased.

Finally we checked if there are some questions that some systems can answer but are not answerable from the gold annotations. We found that there were 11 such instances for TIPSem system and 12 such instances for TRIOS system.

For example, one such instance is the *yes/no* example we showed in the previous section - `IS e1 AFTER e9`. The gold annotation has the relation `e1 SIMULTANEOUS e4`, but it misses other necessary relations to infer the relationship between e1 and e9. However, one system can answer this particular question. This experiment supports our claim about the incompleteness of human temporal annotations.

The incompleteness of human temporal annotation is also discouraging for evaluating automated systems in corpus-based evaluation. Since we will penalize automated systems for some good extractions. All human annotation-based schemes suffer this problem. In this case, annotating temporal information is much harder than answering temporal questions from text, i.e. there is less chance to have wrong human answers than wrong human annotation. As a result, evaluating temporal understanding with QA compared to corpus-based evaluation is more reliable.

V. CONCLUSION

Motivated by the hypothetical benefits of using QA to assess the understanding of the temporal information, in this paper, we have analyzed QA as evaluation metric. For that purpose, we have developed a QA system capable of reasoning about time and of answering questions about TimeML annotated documents (system and questions available on-line[2]).

Human annotation of temporal information can not be avoided since it is needed for training and developing temporal information extraction systems. Nevertheless, for evaluation, in this paper we have shown the advantages that QA offers in contrast to corpus-based evaluation. These are (i) creating question-answer sets about the documents is less time consuming for humans – creating questions only requires reading and understanding the text, and ask simple or complex questions about it, while temporal annotation requires training of the annotators and TimeML involves the annotation of many elements; and (ii) incorrectly answering a temporal question about a document is less probable for humans. QA is, therefore, better to rapidly obtain larger and more reliable performance tests for temporal understanding.

The results obtained have also shown that: (i) reasoning is needed to perform temporal QA; and (ii) current TimeML gold standard annotations are not complete enough, since some questions about the documents, which can be answered by humans, cannot be answered from gold annotations.

As future work, we want to perform a large-scale experiment by gathering questions from Amazon mechanical turk[3] or other similar crowd services. Furthermore, we aim to use our QA evaluation to analyze the performance of automated temporal information understanding systems in new domains.

REFERENCES

[1] J. Pustejovsky, J. M. Castao, R. Ingria, R. Sauri, R. J. Gaizauskas, A. Setzer, G. Katz, and D. R. Radev, "TimeML: Robust Specification of Event and Temporal Expressions in Text." in *New Directions in Question Answering*, M. T. Maybury, Ed. AAAI Press, 2003, pp. 28–34.

[2] M. Verhagen, R. Sauri, T. Caselli, and J. Pustejovsky, "Semeval-2010 task 13: Tempeval 2," in *Proceedings of International Workshop on Semantic Evaluations (SemEval 2010)*, 2010.

[3] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communication ACM*, vol. 26, no. 11, pp. 832–843, 1983.

[4] J. Pustejovsky, J. Wiebe, and M. Maybury, "Multiple-perspective and temporal question answering," in *Proceedings of the Proceedings of LREC workshop on question answering: Strategy and Resources*, 2002.

[5] J. Hobbs and J. Pustejovsky, "Annotating and reasoning about time and events," in *Proceedings of the AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, 2003.

[6] S. Harabagiu and C. A. Bejan, "Question answering based on temporal inference." in *Proceedings of the AAAI-2005 Workshop on Inference for Textual Question Answering*, 2005.

[7] D. Moldovan, C. Clark, and S. Harabagiu, "Temporal context representation and reasoning," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

[8] N. Chambers and D. Jurafsky, "Jointly combining implicit constraints improves temporal ordering," in *EMNLP-08*, Palo Alto, CA, USA, 2008.

[9] E. Brill, J. Lin, M. Banko, S. Dumais, and A. Ng, "Data-intensive question answering," in *Proceedings of the 10th TREC*, 2001.

[10] E. Saquete, J. L. Vicedo, P. Martnez-Barco, R. Muoz, and H. Llorens, "Enhancing qa systems with complex temporal question processing capabilities," *Journal of Artificial Intelligence Research*, 2009.

[11] S. Miller and L. K. Schubert, "Time revisited," in *Proceedings of Computational Intelligence, 6(2)*, 1990.

[12] N. UzZaman and J. Allen, "Temporal evaluation," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.

[13] H. Llorens, E. Saquete, and B. Navarro, "Tipsem (english and spanish): Evaluating crfs and semantic roles in tempeval-2," in *Proceedings of the International Workshop on Semantic Evaluations (SemEval-2010), Association for Computational Linguistics (ACL)*, 2010.

[14] N. UzZaman and J. Allen, "Trips and trios system for tempeval-2: Extracting temporal information from text," in *Proceedings of the International Workshop on Semantic Evaluations (SemEval-2010), Association for Computational Linguistics (ACL)*, 2010.

[2] http://www.cs.rochester.edu/u/naushad/temporal

[3] mturk.amazon.com