An Answer Bank for Temporal Inference

Sanda Harabagiu and Cosmin Adrian Bejan

Human Language Technology Research Institute
The University of Texas at Dallas
Richardson, TX 75083-0688, USA
sanda,ady@hlt.utdallas.edu

Abstract

Answering questions that ask about temporal information involves several forms of inference. In order to develop question answering capabilities that benefit from temporal inference, we believe that a large corpus of questions and answers that are discovered based on temporal information should be available. This paper describes our methodology for creating AnswerTime-Bank, a large corpus of questions and answers on which Question Answering systems can operate using complex temporal inference.

1. Introduction

TimeML (Hobbs and Pustejovsky, 2003) is a corpus annotated with: (a) time expressions; (b) events and (c) links between them. These annotations enable several forms of temporal inference (Boguraev and Ando, 2005), (Moldovan et al., 2005), (Harabagiu and Bejan, 2005). However, additional forms of temporal inference are involved when answering questions. For example, in TimeML, the passage illustrated in Figure 1 has annotations that relate (a) the temporal expression *"May 22, 1995"* to the verb phrase *"made a brigadier general"* and (b) the temporal expression *"the following year"* to the verb phrase *"appointed military attache"*. This passage is answering the question *"Q_1: How long it took Frakas to become military attache at the Hungarian embassy in Washington after his promotion to brigadier general ?"*

> On May 22, 1995, Frakas was made a brigadier general, and the following year he was appointed military attache at the Hungarian embassy in Wa–shington.

Figure 1: Example of passage from TimeML.

Automatic Question Answering (Q/A) involves (1) the question processing; (2) the passage retrieval; and (3) the answer extraction. When processing question Q_1, three goals must be achieved:

GOAL 1: As reported in (Harabagiu et al., 2001) the *expected answer type* (EAT) of the question must be determined. In the case of Q_1, the EAT is a TIME_DURATION. This EAT is typically associated with question stems of the form *"How long"* and with idiomatic expressions like *"it takes"*.

GOAL 2: Second, question processing involves the discovery of dependencies between the EAT and the other concepts from the question. When we apply shallow semantic parsing on Q_1, we discover the dependencies illustrated in Figure 2. The semantic information is produced by a semantic parser trained on the PropBank annotations (*www.cis.upenn.edu/~ace*), which was reported in (Moschitti and Bejan, 2004). The semantic parser is able to recognize predicate-argument structures in which the predicates are lexicalized by (a) verb or (b) nominalizations. For the case when predicates are nominalizations, the seman-
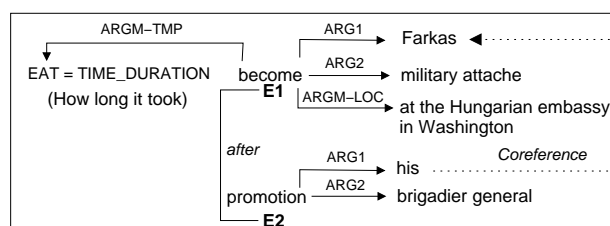

Figure 2: Temporal and semantic dependencies in a question.

tic parser relies on its classifiers trained on the NomBank annotations (*http://nlp.cs.nyu.edu/meyers/NomBank.html*). For example, in Figure 2, the predicate-argument structure E1 is generated due to the data available from PropBank, whereas the recognition of E2 is enabled by data available from NomBank. Furthermore, the two predicate-argument structures are connected by a *temporal relation* made explicit by the signal *"after"*. This dependency needs to be interpreted as: (i) the beginning of the time duration sought by the EAT is simultaneous with the event illustrated as E1 in Figure 2 and (ii) the end of the time duration sought by the EAT is simultaneous with the event illustrated as E2 in Figure 2.

GOAL 3: Keywords from the question need to be selected. The semantic dependencies resulting from the fulfillment of GOAL 2 help selecting the best keywords. The keywords are grouped in two classes, each corresponding to a different predicate-argument structure that needs to be retrieved. The first class of keywords KC_1 includes $KC_1=\{K_1="Farkas", K_2="military", K_3="attache", K_4="Hungarian", K_5="embassy", K_6="Washington"\}$, whereas $KC_2=\{K'_1="Farkas", K'_2="brigadier", K'_3="general"\}$. Moreover, the keywords and the EAT are expected to establish meaningful semantic relations in the passages that are retrieved.

The passage retrieval module for our Q/A system is using the keyword classes to express semantic constraints that are expected to be met by the relevant passages. Some of the semantic constraints are using temporal inference. The two queries that are generated based on KC_1 and KC_2 are:

QUERY$_1$=[ARG1(K_1), ARG2(K_2, K_3), ARGM-LOC(K_4, K_5, K_6), ARGM-TMP(END(EAT))]

QUERY$_2$=[ARG1(K'_1), ARG2(K'_2, K'_3), ARGM-TMP (START(EAT))]

The passage retrieval component of our Q/A system returns a ranked list of passages to each of the queries. The answer extraction module needs to select the partial answers and to infer the correct answer. If it selects the passage illustrated in Figure 1, the answer is *"around one year"*. In the passage, the time duration is not explicit, but a temporal expression is linked to each of the events. However, two more problems hinder the answer inference process: (1) the events from the question do not match the events from the passage, thus the confidence that they are paraphrases needs to be assessed; and (2) there is no temporal signal like *"after"* connecting the two events in the passage, thus other form of temporal inference needs to be used. The first problem is addressed by acquiring paraphrases of events, whereas the second problem is solved by having access to *temporal normalizations*. For example, the normalization of temporal expression TE_1=*"the following year"* from the passage illustrated in Figure 1, is 1996DDMM (where DD represent the day of the MM, which is the month), because the reference to the implicit current year is resolved to 1995, which was derived from TE_2=*"May 22, 1995"*. The two temporal expressions have the roles TE_1=END(EAT(Q_1)) and TE_2=START(EAT(Q_1)). When computing the TIME_DURATION from the normalizations of expressions TE_1 and TE_2, the answer extractor cannot generate an exact answer, but only the approximation *"around one year"*. This is because of the unknown month and day from the normalization of TE_1. If the MM digits are between 01 and 05 the TIME_DURATION is less than a year, whereas if it is larger than 05, it becomes more than a year.

To enable Question Answering systems to operate with complex temporal inference, there is need of a large corpus of questions and answers on which Q/A systems can be trained. We created such a corpus, that we call **AnswerTime-Bank**, in which the answers are selected and benefit from the TimeML annotations. We aimed at producing a large set of complex questions, that are answered by different forms of temporal inference. (Saquete et al., 2004) has illustrated the need for such resources. Our annotations mark: temporal normalizations, paraphrases, as well as inference that justifies the answer.

Additionally, temporal inference interacts with other forms of textual inference, that may benefit the Q/A task. The recent PASCAL RTE evaluation (Dagan et al., 2005) as well as the AQUAINT inference evaluations have shown need for capabilities to infer and draw entailments constrained by temporal information. Textual entailment has been defined as the task of deciding, given two text fragments, whether the meaning of one of the texts can be inferred from the other text. The AQUAINT KB evaluations have also considered the case when one of the texts is a question, the other text is a background to the question, and the textual inference enables the answering to the question. For example, Figure 3 illustrates the question Q_{KB} that is entailed by the passage P_{KB} because the prediction of a further increase presupposes a past increase.

To be able to infer the answer A_{KB}, we need to recognize:

1. Two events in P_{KB}: e1 = the predicting event and e2 = the increasing event, in which e2 is temporally constrained to

---

P_{KB} : The Russian Emergencies Ministry predicts a further increase in 1999 of the concentration of the toxic agents in marine burials of chemical weapons.

Q_{KB} : Has the concentration of toxic agents in marine burials of chemical weapons increased prior to 1999?

A_{KB} : Yes.

Figure 3: Answering temporal questions with entailment.

happen DURING 1999;

2. The event e2 = the increasing event in Q_{KB} which this time is constrained to happen BEFORE 1999;

3. The factive relation between event e1 and e2 in P_{KB}; and most importantly

4. The interpretation of the modifier *'further'* for event e2, which indicates that there is a CONTINUATION of e2 from a previous time.

Based on this information, the answer A_{KB} may be inferred. Figure 4(a) illustrates the events, modifiers and temporal expressions from Figure 3. We represent events as
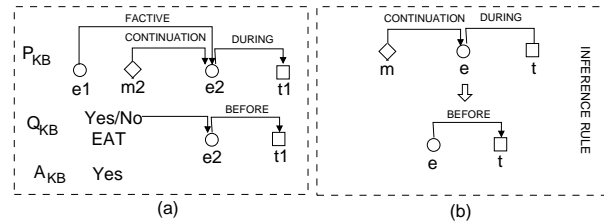

Figure 4: Inference rule that enables the entailment from Figure 3.

circles, their modifiers as diamonds and temporal expressions as squares. The EAT is represented as well. For example, if the modifier m indicates that the event e shall be in a CONTINUATION process, and the event e takes place DURING the time period t, we infer that the event e was also happening before the time period t. Inference rules, like the one illustrated in Figure 4(b), are based on possible relations that exist between (a) events; (b) time expressions and (c) modifiers of events. Example of such temporal relations were introduced in (Allen, 1991). Temporal relations, when discovered, may lead to other questions than Q_{KB} which was illustrated in Figure 3. Two examples of additional questions that are answered by P_{KB} are:

Q_{KB}^1 : What did the Russian Emergencies Ministry predict in 1999 ?

Q_{KB}^2 : When did the increase of the concetration of the toxic agents in marine burials of chemical weapons happen ?

All these questions and their answers are useful for Q/A system developers. Question Q_{KB} tests the ability to use temporal inference, whereas question Q_{KB}^1 or Q_{KB}^2 test the ability to locate information that is constrained temporally.

The reminder of the paper is organized as follows. Section 2 describes the methodology employed for selecting questions and answers in our TimeAnswer-Bank. Section 3 details the bootstrapping of new data. Section 4 reports on the usage of semantic and pragmatic knowledge required by temporal inference in Q/A. Section 5 summarizes the conclusions.

2. Question and Answer Selection Based on TimeML Annotations

Time expressions anchor events and states in narratives. They do the same anchoring in questions. We have used human-generated questions and annotated them in the same way as narratives are annotated in TimeML. There are three types of objects that are annotated:

- **Time expressions**, annotated through TIMEX3 tags;
- **Event expressions**, corresponding to EVENT tags;
- **LINK tags** that encode various relations that hold between temporal elements.

There are three types of TIMEX3 expressions: (a) fully specified temporal expressions, e.g. *"August 14, 1990"*; (b) underspecified temporal expressions, e.g. *"Monday"*, *"next month"*, *"last year"*, *"two days ago"*; and (c) durations, e.g. *"two months"*, *"a week"*. In addition, a TIMEX3 expression can provide a temporal anchor for other temporal expressions in the document.
In TimeML, seven types of events are considered:

◁ 1 ▷ *occurence, e.g. "die", "crash".*
◁ 2 ▷ *state, e.g. "on board", "kidnapped", "loved".*
◁ 3 ▷ *reporting, e.g. "say", "report", "announce".*
◁ 4 ▷ *immediate–action, e.g. "attempt", "try".*
◁ 5 ▷ *immediate–state, e.g. "believe", "intend".*
◁ 6 ▷ *aspectual, e.g. "begin", "finish", "stop".*
◁ 7 ▷ *perception, e.g. "see", "hear", "feel".*

In TimeML texts, there are annotations of two types of relations:

- binary relations, that are established between (i) pairs of events or (ii) events and temporal expressions; and
- signaled relations, which link events and/or temporal expressions through *temporal signals*.

Temporal signals are: (a) temporal prepositions, e.g. *"during"*, *"on"*, (b) temporal connectors, e.g. *"when"*, *"while"* and (c) temporal subordinates, e.g. *"if"*, *"then"*.
To capture all temporal relations in text and to provide means for disambiguating them, TimeML uses a set of three LINK tags:

1 ▷ TLink or Temporal Link[1], representing temporal relations holding between events or between an event and a time;

2 ▷ SLink or Subordination Link[2], used for contexts introducing relations between two events; and

3 ▷ ALink or Aspectual Link[3] representing the relationship between an aspectual event and its argument event.

Additionally, we have marked up modifiers that entail temporal information, similarly to the adjective *"further"* in Figure 3. We have used a new LINK tag, that we called MLink, for Modifier Link. The relations made explicit by MLink overlap with relations made explicit by TLink, SLink and ALink.

---

[1]The TLink makes explicit the following relations: (1) Before; (2) After; (3) Includes; (4) Is Included; (5) During; (6) Simultaneous; (7) Immediately After; (8) Immediately Before; (9) Identity; (10) Begins; (11) Ends; (12) Begun By and (13) Ended By.

[2]The SLinks are one of the following sorts: (1) Modal; (2) Negative; (3) Evidential; (4) Negative Evidential; (5) Factive; (6) Counter-Factive and (7) Conditional.

[3]The ALink relations are (1) Initiates; (2) Culminates; (3) Terminates; (4) Continues and (5) Reinitiates.

The annotations available from TimeML can be used for selecting answers for which we can generate multiple questions. In Section 1 we have exemplified an answer originating in TimeML (Figure 1) and we have discussed how it can answer question Q_1. Our search for answers available from TimeML starts with the discovery of two temporal expressions T_1 and T_2. The *Answer Selection Procedure* is:

Step 1: Discover T1 and T2, temporal expressions in the same sentence or in adjacent sentences.
Step 2: Find events E1 and E2 linked to T1 and T2 respectively.
Step 3: Find link chains CE1 between E1 and other events.
Step 4: Find link chains CE2 between E2 and other events.
Step 5: Use implicit temporal inference on CE1 and CE2.

When applying the *Answer Selection Procedure* to the example illustrated in Figure 5 we discover: (1) temporal expressions t1 and t2 (Step 1); (2) events e1 and e4 linked to t1 and t2 with TLink:Is_Included (Step 2); and (3) the event chain {e1, e2, e3} (Step 3). Because t1 and t2 are linked (by an AnchorTime(t2) = t1) we conclude that {e1, e2, e3} and e4 are simultaneous (Step 5).
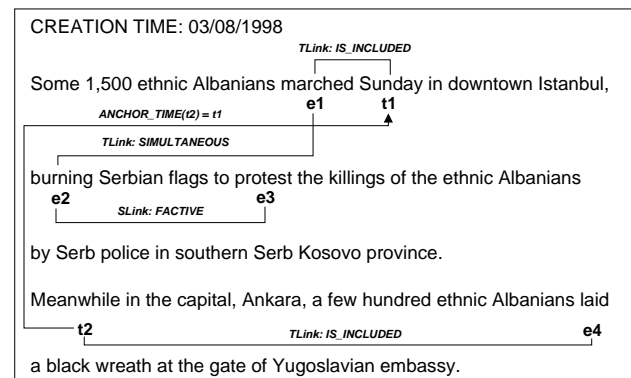

Figure 5: Example of TimeML annotation.

Figure 6 illustrates three forms of temporal inference that are dictated by the types of links in event chains. In Figure 6(a), the fact that the anchor of t2 is t1 indicates that events e1 and e2 must be simultaneous. Therefore, the conclusion of the temporal inference rule illustrated in Figure 6(a) creates a new TLink:Simultaneous relation between the two events. In Figure 6(b), the event chain created from e1 {e1, e3} has a TLink:Before, indicating that e3 happened before e1. If the anchor of t2 is t1, the temporal inference rule has two conclusions: (1) that between e3 and e2 there is a TLink:Before and (b) that between e1 and e2 there is a TLink:Simultaneous. For the example illustrated in Figure 5, the temporal inference rule that applies (Step5 of *Answer Selection Procedure*) is illustrated in Figure 6(c). There are three conclusions of the temporal inference rule illustrated in Figure 6(c) because there were three events in the event chain connected to t1 and only one event connected to t2.
Figure 6 illustrates the format of our implicit temporal inference rules. The left-hand side of the rule represents the possible relations between events (chains) and temporal expressions whereas the right-hand side represents one or more conclusions which are expressed by pairs of events connected by new TLink expressions.
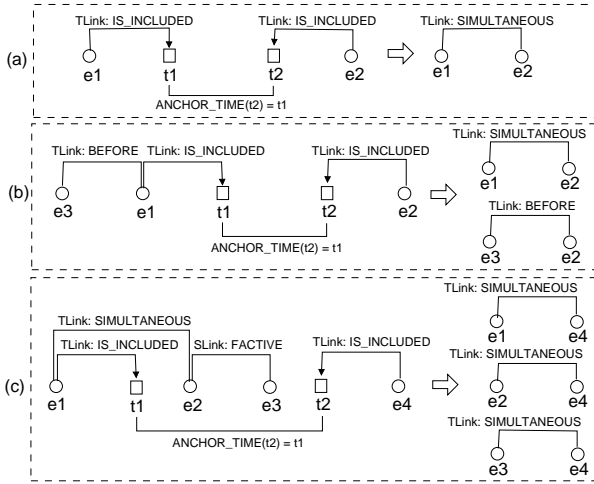
Figure 6: Inference rules based on TimeML links.

When the answer is selected and implicit inference has been discovered, we can generate questions that require temporal inference. For the answer illustrated in Figure 5, since all events are simultaneous (as indicated by the implicit inference rule from Figure 6(c)), we can refer to all events with the generic expression (e.g. *"actions"*) on a specific time (e.g. *"Sunday March 8, 1998"*). Furthermore, the predicate-argument structures derived from the two sentences illustrated in Figure 5 indicate that all events have as actors ethnic Albanians. Thus, we may associate this paragraph with the generic question Q_1^G.

> Q_1^G: What actions were taken by ethnic Albanians in Turkey, on Sunday, March 8, 1998 ?

In order to create the AnswerTime-Bank, we also need a *Question Suggestion Procedure* which employs (a) the answers selected as well as (b) the forms of temporal inference that are available on them. This procedure also uses 40 different possible EATs to produce question suggestions. The *Question Suggestion Procedure* is:

> *Step 1:* Find EATs compatible with the selected answers and place them in [All–EATs]
> *Step 2:* For every EAT from [All–EATs] *(Begin loop)*
> *Step 3:* Use the semantic dependencies from the answer to suggest the question dependencies.
> *Step 4:* Map the dependencies on a set of question patterns.
> *Step 5:* Ask the linguist researcher to suggest a question by using a paraphrase having the same semantic dependencies.
> *Step 6:* Validate the question. *(End loop)*

For example, the EAT of Q_1^G is a list of actions carried by the same agent:*"ethnic Albanians"* and in the same location:*"Turkey"* on the same date:*"March 8, 1998"*. Also, the factive relationship between e2:*"burning"* and e3:*"protest"* indicates that the actions that were referred to in Q_1^G can be specialized, as *"forms of protest"* and enable the generation of Q_2^G. The other questions that were generated had either the time as the expected answer (Q_3^G) or some of the entities involved in the events constrained by time (for example Q_4^G, Q_5^G). When between two events we find an SLink:FACTIVE relation, since such relations introduce a presupposition or entailment between the events, we can generate a question that requests causal information (Q_6^G). The questions Q_1^G, Q_2^G, Q_3^G, Q_4^G, Q_5^G and Q_6^G, illustrated in Figure 7, were created by humans such that

Q/A system developers can test their ability to answer them when employing (i) textual inference and (ii) relations between events and temporal expressions. Not all questions that humans generated were factual and related to a single date. For example, for the passage illustrated in Figure 1, we generated the question Q_1, introduced in section 1, which asks about a time interval that is not explicit in the passage.

To be able to create a TimeAnswer-Bank that encodes a large variety of questions that require temporal inference we needed to recognize automatically the temporal expressions, events and their interconnecting links such that we could find many examples that use the same form of inference. With the annotations from TimeML, we were able to detect 4125 answers, to which we applied 120 implicit temporal inference rules similar to those illustrated in Figure 6. Because we found that event chains can have lengths from 1 to 6, we believed that it would be useful to have all the possible combinations of such links available such that we can generate questions that exploit the implicit temporal inference. In the first phase of our work we have used event chains with the maximum length of 4.

> Q_2^G: What forms of protest took place in Turkey, on Sunday, March 8, 1998?
> Q_3^G: When did the ethnic Albanians protest in Turkey, in March 1998?
> Q_4^G: Who were the protesters in Turkey, in 1998?
> Q_5^G: How many Albanians protested in Turkey, in March 1998?
> Q_6^G: Why did the ethnic Albanians protest in Turkey, in March 1998?

Figure 7: Example of questions generated for the text illustrated in Figure 5.

3. Bootstrapping the AnswerTime-Bank

The *Answer Selection Procedure*, together with the *Question Suggestion Procedure*, enabled us to assemble 3472 questions that require temporal inference and to have available answers for them as well as annotations that inform the temporal inference. However, in this form, AnswerTime-Bank has several limitations . First, we could not assemble examples for all the forms of questions that require temporal inference that were listed in (Harabagiu and Bejan, 2005). Second, for each type of question, we did not have a very large number of examples. Third, due to the limitations of the *Answer Selection Procedure*, we did not have any instance of answers that originated in different documents. In order to address these issues, we have started to bootstrap the AnswerTime-Bank by selecting answers from the AQUAINT corpus. In the bootstrapping procedure, we have modified the *Answer Selection* such that the pair of time expressions do not necessarily belong to the same or adjacent sentences. The bootstrapping procedure requires the discovery of (1) time expressions; (2) events; (3) temporal signals and (4) links between them. To discover time expressions, we relied on the TIMEX3 annotations produced for us by the TASER time recognition and normalization system (Aarseth et al., 2005). We considered as events only the verbs, which are part of predicate-argument structures recognized by our semantic parser (Moschitti and Bejan, 2004), filtering out all the forms of the verb *"be"* and several form of generics as well, as is described in (Sauri et

al., 2006). To classify events in text we implemented similar methods as the ones described in (Sauri et al., 2005). Temporal signals were recognized based on lexicons. We also needed to discover the three types of links. For this reason, we have developed and implemented four link detection methods that are illustrated in Figures 8, 9, 10, 11. Since TLink relations need to be identified in the AQUAINT corpus, we have implemented a method for automatically recognizing such relations by extending the method reported in (Lapata and Lascarides, 2004), which aimed the discovery of temporal constraints between two clauses from the same sentence.

Predicate-argument structures discovered by semantic parsers enable us to detect relations between events expressed as verbs and temporal expressions. But such predicate-argument structures do not indicate what type of TLink exist. Thus, we first generated a classifier, of which features are illustrated in Figure 8, that enabled us to detect TLinks between such events and temporal expressions.

```
TLink Detection Method 1
  Input:  – Predicate–argument structure with ARGM–TMP
          – TIMEX3 expression
  Output: – TLink (Y/N) and TLink class

We have trained a decision tree classifier that considers the
following features:
  –verb lemma
  –verb tense
  –the temporal signal that begins the ARGM–TMP (if it exists)
  –the temporal signal that ends the ARGM–TMP (if it exists)
  –the temporal signals that are in the clause containing the verb
  –distance in words between ARG–TMP and the verb
  –position of the ARG–TMP with respect to the verb
  –presence in ARGM–TMP of words like: earlier, next, last
  later, past, future, recently, late, previously, over, ago
```

Figure 8: Method 1 for discovering TLink relations.

For discovering temporal relations between events in free text, we used an event graph-based representation. Specifically, the nodes in the graph are represented by events and the edges between the nodes are either TLink, SLink or ALink relations. We have extended the model proposed in (Lapata and Lascarides, 2004) for classifying the TLink relations between events in two consecutive sentences and we also have enhanced the model with additional features. Concretely, for each pair of events from the same sentence or from consecutive sentences we used an SVM classifier that predicts and classifies a possible TLink relation. The features used for training the classifier are illustrated in Figure 9. For discovering TLink relations at the discourse level, we observed that transitional words introducing sentences or clauses play an important role. For example, transitional words expressing *addition* like *"in addition"*, *"additionally"*, *"moreover"* introduce SIMULTANEOUS TLink relations, *result* transitional words like *"as a result of"*, *"in consequence"* introduce AFTER and BEFORE relations, while *time* transitional words like *"meanwhile"*, *"immediately"*, *"in the meantime"*, *"in the past"*, *"in the future"*, *"finally"*, *"then"*, *"next"*, *"afterward"* may introduce all the types of TLink relations.

All these TLink relations represent the edges in the event graph built over the entire text for which the method is applied. However, we cannot rely entirely on the method presented above and therefore we have to check the con-

```
TLink Detection Method 2
  Input:  – a pair of events in the same sentence or in two con–
          secutive sentences
  Output: – TLink (Y/N) and TLink class

We have trained an SVM classifier that considers the
following features:
  –all the features described in (Lapata and Lascarides, 2004)
  since they perform well in discovering temporal relations
  between events in the same sentence
  –the temporal signals between the two verbs
  –the temporal signals that are in the clauses containing the verbs
  –distance in words between the two verbs
  –transitional words introducing sentences or clauses of the verbs
```

Figure 9: Method 2 for discovering TLink relations.

sistency of the event graph and to remove all contradictory relations between two events in the graph. For this, we inferred all the possible temporal relations between two events in the graph following all possible paths that connect these two events. If we find contradictions in the inferred temporal relations, we discard all the temporal relations that connect these two events. An example of a contradiction in an event graph is: if we have event E_1 TLink:AFTER event E_2 and E_2 TLink:SIMULTANEOUS E_3, then we cannot have in the event graph E_1 TLink:BEFORE E_3. We also discard all the TLink relations in the event graph that can be replaced by an ALink or SLink relation discovered by the next two methods.

ALink relations represent the temporal relations introduced by aspectual events. We observed in TimeML corpus that different aspectual events trigger different types of aspectual relations. For example, the most frequent aspectual events for each type of the ALink relation in TimeML are:

- *initiation*: *"open"*, *"begin"*, *"become"*, *"start"*, *"trigger"*.
- *termination*: *"end"*, *"suspend"*, *"stop"*, *"abandon"*.
- *continuation*: *"extend"*, *"reinsate"*, *"remain"*, *"continue"*.
- *reinitiation*: *"resume"*, *"restore"*, *"return"*.
- *culmination*: *"finish"*, *"complete"*, *"reach"*.

Starting from this observation, we derived the method illustrated in Figure 10 that identify aspectual relations.

```
ALink Detection Method
  Input:  – a pair of events in the same sentence
  Output: – ALink (Y/N) and ALink class

The method for identifying aspectual relations is described in
the following steps:
1. Build aspectual event clusters from TimeBank with the most
frequent events that introduce ALink relations.

2. Bootstrap the clusters with aspectual events that require se–
mantic processing. To accomplish this task, we used WordNet
relations for determining if a event is in relation with events from
the aspectual event clusters constructed at Step 1. For exam–
ple, we classify "graduate" as an event that introduce culmination
relation, because it has "culminates" in its WordNet gloss.

3. Identify an ALink relation between two events inside a sentence
if: (a) the first event is an aspectual event that belongs in one of
the five aspectual event clusters and (b) the second event is situa–
ted in the same verbal phrase structure with the first event. Label
relation with the cluster label of the aspectual event.
```

Figure 10: Method for discovering ALink relations.

In general, the SLink relations are introduced by particular classes of events. Some of these classes are presented below:

- *events expressing presuppositions and beliefs*: *"think"*, *"believe"*, *"try"*, *"predict"*, *"want"*, *"able to"*, *"hope"*.
- *perception events*: *"see"*, *"look"*, *"hear"*, *"perceive"*.
- *reporting events*: *"say"*, *"tell"*, *"report"*, *"quote"*.
- *events expressing negative polarity*: *"deny"*, *"reject"*.

We build these semantic classes of events form TimeML and, in a similar way as in ALink method, we used Word-Net to enrich the semantic classes with additional events. Not only this classification of events help in identifying the SLink relations, but also they are used as features in a multiclass classifier for identifying the SLink relation types as illustrated in Figure 11. For example, reporting and perception events introduce EVIDENTIAL SLink relations and events expressing negative polarity introduce NEGATIVE EVIDENTIAL SLink relations. Other features we used for classifying the SLink relations are illustrated in Figure 11.

```
SLink Detection Method

   Input:   – a pair of events in the same sentence
   Output: – SLink (Y/N) and SLink class

We have trained an SVM classifier that considers the
following features:
–verb lemma
–verb tense
–lexical features and the part of speeches for the words
surrounding the two events
–the semantic class of the events (if the case)
–the type of the events
–the temporal signals that are present in the event clauses
–obligation: presence of modals like "must", "ought", "should"
–possibility: presence of modals like "can", "could"
–ability: presence of modals like "might", "may"
–future: presence of modals like "will", "shall", "would"
–negation: presence of words like "not", "n't"
```

Figure 11: Method for discovering SLink relations.

Many complex questions do not have the entire answer in the same document; they require answer fusion. In view of this condition, we have included new documents, annotated them in the same way as TimeML and then decided on the partial answers before creating the complex question. Our resource characterizes both question decomposition and the answer fusion in terms of types of links between events or events and time expressions. One key aspect of the bootstrapping process is the identification of answer types for questions created by humans. They enable us to propose new questions and answers. For example, given an answer type FORMS-OF-PROTEST that is constrained by a given date (for Q_2^G), we acquired a set of patterns that represent forms of protest with the method reported in (Thelen and Riloff, 2002) and determined which events occurred in the same time and location. Then we replaced the date to generate questions like Q_7^G.

Q_7^G: Who were the protesters during the Scotland Summit in 2004?

This is an example of complex question, where we employed the temporal connector *"during"* to express the temporal constrains.

4. Inference with Semantic and Pragmatic Knowledge

One important property of the AnswerTime-Bank is the semantic and pragmatic variation between questions and answers. We have carefully used (1) paraphrases of the answer and (2) generalizations such that we could allow for semantic and pragmatic inference while processing temporal questions. Consequently, we have also annotated the forms of semantic knowledge that are required and suggested possible sources of such knowledge. For example, often domain knowledge was required. For the sentence,

In fiscal 1989, Elco earned $7.8 million, or $1.65 a share.
to produce the question
What was Elco's revenue in 1989?
we relied on semantic glossing of the concept *"revenue"* as the money earned by a company during a given year. Such a gloss is available from WordNet, and we have encoded the mappings between the question and answer concepts using the WordNet glosses. An important factor is the bootstrapping of such lexico-semantic resources that account for the inference of temporal answers. The resource is important as well for studying paraphrases under temporal constraints.

5. Conclusions

We have described the methodology we employed to date for generating a corpus of questions and answers that require temporal inference. AnswerTime-Bank was built using the annotations from TimeBank. We have described as well our method of bootstrapping the resource by discovering automatically TimeBank-like expressions and links. We believe that AnswerTime-Bank shall be a valuable resource for researchers interested in Question Answering.

Acknowledgments

6. References

Paul Aarseth, Murat Deligonul, John Lehmann, Luke Nezda, and Andrew Hickl. 2005. ACE 2005 TERN System Description: TASER. In *Proceedings of ACE 2005 Workshop*.

James F. Allen. 1991. Time and time again: The many ways to represent time. *International Journal of Intelligent Systems*, 6(4):341–356.

Branimir Boguraev and Rie Kubota Ando. 2005. TimeML-Compliant Text Analysis for Temporal Reasoning. In *Proceedings of the Nineteenth International Joint Conference on Artifi cial Intelligence (IJCAI-2005)*.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The PASCAL recognizing textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop Recognizing Textual Entailment*.

Sanda Harabagiu and Cosmin Adrian Bejan. 2005. Question Answering Based on Temporal Inference. In *Proceedings of the AAAI-2005 Workshop on Inference for Textual Question Answering*.

Sanda M. Harabagiu, Dan I. Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan C. Bunescu, Roxana Girju, Vasile Rus, and Paul Morarescu. 2001. The Role of Lexico-Semantic Feedback in Open-Domain Textual Question-Answering. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*, pages 274–281.

Jerry Hobbs and James Pustejovsky. 2003. Annotating and Reasoning about Time and Events. In *Proceedings of the AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*.

Mirella Lapata and Alex Lascarides. 2004. Inferring Sentence-internal Temporal Relations. In *In Proceedings of the North American Chapter of the Assocation of Computational Linguistics, 153-160. Boston*.

Dan Moldovan, Christine Clark, and Sanda Harabagiu. 2005. Temporal Context Representation and Reasoning. In *Proceedings of the Nineteenth International Joint Conference on Artifi cial Intelligence (IJCAI-2005)*.

Alessandro Moschitti and Cosmin Adrian Bejan. 2004. A Semantic Kernel for Predicate Argument Classification. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, Boston, MA,USA.

E. Saquete, P. Martínez-Barco, R. Munoz, and J.L. Vicedo. 2004. Splitting Complex Temporal Questions for Question Answering systems. In *Proceedings of the 42th Annual Conference of the Association of the Association for Computational Linguistics (ACL-04)*, pages 567–574.

Roser Sauri, Robert Knippen, Marc Verhagen, and James Pustejovsky. 2005. Evita: A Robust Event Recognizer for QA Systems. In *Proceedings of HLT/EMNLP*.

Roser Sauri, Jessica Littman, Bob Knippen, Robert Gaizauskas, Andrea Setzer, and James Pustejovsky. 2006. TimeML Annotation Guidelines. http://www.timeml.org.

Michael Thelen and Ellen Riloff. 2002. A Bootstrapping Method for Learning Semantic Lexicons Using Extraction Pattern Contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*.