RC23892 (W0602-203) February 27, 2006
Computer Science

IBM Research Report

Analysis of TimeBank as a Resource for TimeML Parsing

Branimir Boguraev, Rie Kubota Ando
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Analysis of TimeBank as a Resource for TimeML Parsing

Branimir Boguraev and Rie Kubota Ando

IBM T.J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598
bran@us.ibm.com, rie1@us.ibm.com

Abstract

In our work, we present an analysis of the TimeBank corpus—the only available reference sample of TimeML-compliant annotation—from the point of view of its utility as a training resource for developing automated TimeML annotators. We are encouraged by experimental results indicative of the potential of TimeBank; at the same time, closer inspection of causes for some systematic errors shows off certain deficiencies in the corpus, primarily to do with small size and inconsistent annotation. Our analysis suggests that even a reference resource, developed outside of a rigorous process of training corpus design and creation, can be extremely valuable for training and development purposes. The analysis also highlights areas of correction and improvement for evolving the current reference corpus into a community infrastructure resource.

1 INTRODUCTION

The primary focus of this investigation[1] is the study of the characteristics of the TimeBank corpus (Pustejovsky et al., 2003b) which are intimately connected to its utility as a training resource for developing automatic TimeML analysis machinery.

TimeML (a Mark-up Language for Time) has been developed as a 'transport mechanism' for temporal information, and it reflects an emerging model of staged temporal analysis where temporal information extraction (IE) from a text document would be followed by a formalisation by means of an ontology of time (Hobbs and Pan, 2004). TimeML (Pustejovsky et al., 2003a) uses the representational principles of XML markup to annotate the analysis of the core elements in a temporal framework: *time expressions*, *events*, and *links* among these (additionally moderated by temporal connectives, or *signals*).

Computational analysis of time is very complex; the complexity arising from the need to facilitate full mapping of temporal links among time expressions and events onto an ontologically-grounded temporal graph (or its equivalent), cf. (Fikes et al., 2003), (Han and Lavie, 2004). TimeML is thus committed to capture all of the temporal characteristics in a text document. Consequently, the language is considerably more expressive in comparison with markup schemes for "named entities" in traditional IE endeavours.

Herein lies the promise of TimeML: in contrast to IE markup practices to date, which target relatively simple phenomena and whose expressive capabilities have not been designed to capture the variety and complexity of information required to support reasoning, TimeML annotations can adequately feed a mapping to a time ontology which, suitably interfaced with an ontology of events, can be used in formal reasoning contexts (Hobbs and Pustejovsky, 2004).

With this promise, however, come challenges. Temporal information extraction—as defined by TimeML's representational properties (as outlined above[2])—is a harder problem than named entity identification alone. Addressing this problem brings to the fore both issues of method and strategy for TimeML-compliant analysis, and questions of infrastructure adequate for such analysis.

In our work, we address both such issues. Elsewhere, we discuss the design, implementation, and performance of an automatic TimeML annotator, deploying a hybrid analytical strategy of mixing aggressive finite-state processing over linguistic annotations with a state-of-the-art machine learning technique capable of leveraging large amounts of unannotated data (Boguraev and Ando, 2005b). Here we will focus primarily on the infrastructure issues.

Our analytical framework leverages the only existing reference corpus annotated within the TimeML annotation guidelines. TimeBank is one of the outcomes of the TERQAS effort (Temporal & Event Recognition for QA Systems; see http://www.timeml.org/terqas/index.html), which over the past 24 months coordinated a series of definitional and follow-up workshops from which emerged the current set of TimeML annotation guidelines. The corpus is the only collection, to date, of "detailed annotations of terms denoting events, temporal expressions, and temporal signals, and, most importantly, of links between them denoting temporal relations" (Pustejovsky et al., 2003b). It is offered primarily as an "empirical basis for future research into the way texts actually express and connect series of events"; additionally, the creators of TimeBank suggest that it could be regarded as a resource for "training and evaluating algorithms which determine event ordering and time-stamping" (ibid.).

At the same time, however, TimeBank was not developed as a training corpus *per se*. It is, in fact, almost a 'side effect' of the TERQAS work: it was largely an exercise in applying the annotation guidelines—as they were being developed—to real texts in order to assess the need for, and then the adequacy of, the language representational devices as they were being designed in the process of TimeML evolution. As such, it was never the subject of rigorous considerations of scope, coverage, size, consistency, double annotation, and inter-annotator agreement.

[1]This work was supported in part by the ARDA NIMD (Novel Intelligence and Massive Data) program PNWD-SW-6059. Portions of this paper were presented at a Dagstuhl Seminar on Annotating and Reasoning with Time.

[2]Some familiarity with TimeML is assumed here. Details of the markup language for time can be found, in particular, in (Saurí et al., 2005).

Still, given that TimeBank is the only reference TimeML corpus in existence, there are certain questions concerning the extent to which it can, in fact, support the development of TimeML-compliant machinery. As long as *some* annotated corpus exists, it will undoubtedly be brought into some training cycle. Indeed, ours is not the only effort in using TimeBank for such a purpose; recently, the TARSQI project has been focusing on developing analysis strategies and heuristics for particular subsets of TimeML components (Verhagen et al., 2005).

In this work, then, we offer an assessment of size and consistency of TimeBank, as we observe that these characteristics of the corpus pose certain challenges to the notion of using it as a training resource. Our findings are informed by the experiences we had in using TimeBank while developing the TimeML annotator.

2 A TIMEML ANNOTATOR

A formulation of the problem of TimeML analysis as an information extraction (IE) task is presented in (Boguraev and Ando, 2005b). We target the full temporal markup language—seeking to extract not only temporal expressions (TIMEX3's), but also EVENTs; and further looking for temporal relations (TLINKs). It is largely the breadth and richness of EVENT and LINK types and instances in text that makes the temporal IE task so challenging.

Our approach crucially relies on using TimeBank as a training resource. The observation that the corpus is very small (by any standard; see Section 3) additionally motivates our strategy to incorporate a learning component (word profiling) specially developed for leveraging large volumes of unlabeled data; this is in addition to using a high-performance classification machinery. The specifics of the task (*e.g.* the particulars of time expression normalisation), the need for rich syntax-derived features, and further considerations of the size of training corpus explain our choice of synergistically deploying finite-state descriptive devices (for TIMEX3 analysis and syntactic mark-up) with machine learning techniques.

(Boguraev and Ando, 2005b) and (Boguraev and Ando, 2005a) present some experimental results illustrative of the performance of the TimeML annotator developed. The experiments are based on modeling some aspects of the task as classification problems, and look at the individual contribution of feature set definition, finite-state machinery, and word profiling technique.

At optimal settings, our results (F-score) are at almost 90% in recognising TIMEX3 expressions[3], and at the low 80-ies in recognising *untyped* EVENTs and TLINKs. These figures drop when *typing* (see Section 3, and (Saurí et al., 2005)) becomes part of the task. While this is directly related to the complexity of the typing of TimeML components, it is also the case that the relatively 'ad-hoc' nature of the TimeBank corpus is at play here: as we pointed out earlier, the fact that TimeBank was not developed under the rigorous process mandated by the production needs of a community-wide reference resource would almost certainly lead to some level of noise in the data.

Thus our results are both indicative of the value of TimeBank as a training resource for TimeML parsing, and the need for an in-depth study into the nature of existing noise—with a view of pointing the way for more infrastructure development work. The next two sections present an analysis of TimeBank as a training resource for TimeML-compliant annotation.

3 QUANTITATIVE ANALYSIS OF TIMEBANK

Practical content analysis of documents relies, broadly, on a variety of 'gisting' approaches, offering surrogate views into what a document is about. Numerous NLP technologies and applications are concerned with identifying text fragments with high information quotient (according to certain task criteria). Typical of such approaches are, for instance, efforts to extract mentions of named entities and broader semantic categories of concepts: in isolation, chained, or linked in relational structures. These trends can be observed in the definition of community-wide efforts like the Message Understanding Conferences (MUC)[4] and the Automatic Content Extraction (ACE) evaluations.[5]. One of the common characteristics of such efforts is that they make, from the outset, infrastructural provisions for the development of a substantial 'reference' corpus, which defines a gold standard ("truth") for the task. The corpus contains materials selected to be representative of the phenomenon of interest; sizes of training and testing samples are carefully considered especially as they depend on the complexity of the task; experienced annotators are used; the corpus is not released until a certain level of inter-annotator agreement is reached. These measures ensure that the reference corpus is of a certain size and quality.

The TimeBank corpus is small. This need not be surprising, given that the TERQAS effort did not commit to producing a 'reference', training-strength, corpus in the sense described above. In fact, TimeBank is almost a 'side effect' of the work: it was largely an exercise in applying the annotation guidelines—as they were being developed—to real texts (news articles, primarily) in order to assess the need for, and then the adequacy of, the language representational devices as they were being designed in the process of TimeML evolution.

The extent to which TimeBank is small is illustrated by the following statistics. The corpus has only 186 documents, with a total of 68.5K words. As there are no separate training and test portions, it would need partitioning somehow; if we held out 10% of the corpus as test data, we have barely over 60K words for training.

To put it into perspective, this is order of magnitude less

[3] A reminder that TIMEX3 is different from, and requires more detailed analysis than, TIMEX2 (itself popularised most recently by the the Time Expression Recognition and Normalization program; see `http://timex2.mitre.org/tern.html`).

[4] See `http://www.itl.nist.gov/iaui/894.02/related_projects/tipster/muc.htm`.

[5] See `http://www.nist.gov/speech/tests/ace/`.

[6] See `http://www.cis.upenn.edu/~treebank/`.

than other standard training corpora in the NLP community: the Penn Treebank corpus[6] for part-of-speech tagging (arguably a simpler task than TimeML component analysis) contains more than 1M words—which makes it over 16 times larger than TimeBank; the CoNLL'03 named entity chunking task[7] is defined by means of a training set with over 200K words. A task closely related to time analysis is ACE's TERN (see footnote 2), which only focuses on TIMEX2. TIMEX3, which extends the TIMEX2 tag (Saurí et al., 2005), is just one of half-a-dozen TimeML components; even so, the TERN training set is almost 800 documents/ 300K words-strong.

Fig. 1 shows a breakdown of the individual TimeML component distributions in the corpus. Overall, the figure of just over 29K counts of temporally-related entities seems to hold some promise, when the task of TimeML analysis is construed, broadly, to be a named entity extraction task. However, the perception quickly shifts as we realise that within the inventory of TimeML tags, only three 'primitive' elements behave like named entities (TIMEX3, SIGNAL, and EVENT): this gives us less than 12K marking (text-consuming) spans in the training data (for 3 different categories, *before* we take into account the problem of associating specific subtypes to these elements; see Fig. 3 below).

The remaining 17.5K TimeML tags in the corpus are non-marking, and require more complex analytical machinery than that of 'vanilla' named entity extraction. The broad categories of INSTANCE and LINK elements reflect a projection of an EVENT token to (an) event INSTANCE, and a relational binding between time expressions and such event instances. Again, broadly speaking, the task is one of relation identification; harder than just named entity extraction.

Viewed from such a perspective, counts of 12K and 17.5K training examples for training entity and relation recognisers, respectively, seem meager. Additionally, we observe that in the particular set of data encapsulated by TimeBank, the derivation of event INSTANCEs from the EVENT tokens is not particularly challenging (non-trivial EVENT to event INSTANCE mapping becomes an issue in the analysis of time frequencies (SETs), of which there are only 7 TIMEX3 annotations in the corpus so typed). Thus, the 8K INSTANCE tags in the corpus contribute almost nothing to the training cycle, and we are left with less than 10K examples of relational (LINK) elements.

TimeML tags	# occurrences: 29331	
	marking	non-marking
Timex3	1423	
Signal	2117	
Event	8243	
Instance		(7966)
ALink		282
SLink		2619
TLink		6681
Total:	11783	(7966) 9582

Fig. 1: Distributions of TimeML components in TimeBank.

[7] See `http://www.cnts.ua.ac.be/conll2003/ner/`.

Fig. 2 gives counts of TIMEX3 classes in TimeBank. It is a highly uneven distribution, with clearly not enough TIME and SET examples. Additionally, adjusting the counts to take account of time expressions found in document metadata (marking, for instance, document creation time, document transmittal time, and so forth)—these are of a very uniform format, and can be found with a trivially simple regular expression pattern—the total number of examples drops to 1245. Again, this is considerably less than TERN's 8K TIMEX2 examples.

TIMEX3 class	# occurrences:
date	975
duration	314
time	80
set	7
Total:	1423
(In document body:)	(1245)

Fig 2: Distribution of TIMEX3 types in TimeBank.

Further illustration of the extreme paucity of positive examples over a range of categories in the TimeBank corpus is shown in Fig. 3 .

TLINK type	# occurrences	EVENT type	# occurrences
IS_INCLUDED	866	OCCURRENCE	4,452
DURING	146	STATE	1,181
ENDS	102	REPORTING	1,010
SIMULTANEOUS	69	I_ACTION	668
ENDED_BY	52	I_STATE	586
AFTER	41	ASPECTUAL	295
BEGINS	37	PERCEPTION	51
BEFORE	35		
INCLUDES	29		
BEGUN_BY	27		
IAFTER	5		
IDENTITY	5		
IBEFORE	1		
Total :	1,451	Total :	8,243

Fig 3: Distribution of (some) types of TimeML components. Note that the count of 1451 TLINKs, while apparently different from the number of 6681 TLINKs reported in Fig. 3, refers only to the TLINKs between an event and a temporal expression, itself in the body of a document. (TLINKs with TIMEX3's in metadata are not counted here.)

The numbers reveal some of the variety and complexity of TimeML annotation: for instance, while Fig. 1 gives counts per component, it is clear that the extensive typing of EVENTs, TIMEX3's and LINKs introduces even more classes in an operational TimeML typology. Thus an event recognition and typing task is, in effect, concerned with partitioning recognised events into 7 categories (a particular implementation of such a partitioning is realised as $(2k+1)$-way classification task, where $k = 7$ in our case). Similarly, for TLINK analysis the relevant comparison is to consider that in contrast to, for instance, the CoNLL'03

named entity recognition task—with training data containing 23K examples of named entities belonging to just 4 categories, TimeBank offers less than 2K examples of TLINKs, which, however, range over 13 category types.

The table additionally shows the highly uneven distribution of both TLINK classes and EVENT types; so much so as to render some of the data in the corpus almost unusable for the purposes of a machine learning framework.

4 QUALITATIVE ANALYSIS OF TIMEBANK

This section makes some observations concerning the types of errors encountered during our analysis of the TimeBank corpus. It is important to emphasise that this is an informal analysis; in particular, there is no quantification of error types. It is equally important to realise that our observations are not intended to be critical of the corpus: as we discuss in Section 1, TimeBank was not instantiated as a reference training corpus, and rigorous processes and controls such as double annotation and inter-annotator agreement were not part of this particular corpus definition cycle.

We are primarily motivated by a desire to understand how to interpret the performance figures presented in the previous section: low numbers are typically indicative of any combination of not enough training data, noisy and inconsistent data, complex phenomenon to be modeled, and inappropriate model(s). Our hope is that by highlighting the kinds of 'natural' errors that a 'casual' (human) annotator tends to introduce into the exercise, a more focused effort to instantiate a larger TimeBank would be able to avoid repetition of these kinds of errors.

There are different types of error, broadly falling into three categories: errors due to failures in the annotation infrastructure, errors resulting from broad interpretation of the guidelines, and errors due to the inherent complexity of the annotation task (possibly compounded by underspecification in the guidelines).

ANNOTATION INFRASTRUCTURE ERRORS

Consider the (excerpt from an) annotated document illustrated in Fig. 4. (For brevity, typing information and additional attributes to TIMEX3 and EVENT tags have been omitted. Apparently an error, most likely in the annotation software, has caused a systematic shift by a single character; the scope of this error is the entire document. Clearly, there is potential for mismatches between the reference annotations above and anything tested against them which has been generated without knowing of this type of error.

```
On th<Timex3>e afternoon of Oct. 1 </Timex3>7, after hours
o<Event>f hagglin</Event>g with five insurance-claims adjusters
over <Event> settlin</Event>g a toxic-waste <Event> sui</Event>t,
four lawyers <Event> ha</Event>d an <Event> agreemen</Event>t in hand.
```

Fig.4: Annotation tool gone wrong.

Equally problematic are situations due to non-linear markup in the corpus: since the TimeML language does not allow for embedded or crossing annotations—like the ones illustrated in Fig. 5—a pre- (or post-) processing cycle (typically carried out within an XML parser process) will likely be thrown off by such malformed XML markup.

```
... <Signal> who <Event> should </Event> </Signal> ...
... <Signal> never <Signal> going </Signal> </Signal> ...
... <Event> lawyers <Signal> went </Signal> </Event> ...
... <Event> the <Signal> settlement </Event> into </Signal> ...
```

Fig.5: Embedded, overlapping, and crossing annotations.

The first three examples are, arguably 'harmless', as there would be no trace of abnormality after simply stripping the tags off. However, the semantics of mutually embedded EVENTs and SIGNALs are clearly dubious, at best. More problematic, of course, is the last example, where crossing brackets would confuse a parser. (As it happened in our case, the XML parser driving the generation of the derived test corpus actually used in the experiments, used to fail silently, causing all remaining annotations in the document, after the point of failure, to be ignored.)

The cause of such errors is most likely a combination of features of the supporting software. It is certainly the case that the examples in Fig. 4 and Fig. 5 illustrate a situation which is no longer true of that software; in particular, following the release of TimeBank, a dedicated effort focused on developing a special purpose annotation tool, designed specifically to address the challenges of producing XML-compliant and internally consistent markup for 'dense' annotation tasks (of which TimeML is a particularly good example) (Pustejovsky et al., 2003c). It is also the case that this problem is not manifested over many documents.

However, TimeBank is sufficiently small so that any additional 'noise' introduced from extraneous sources—even if relatively few documents are impacted—has a noticeable effect on performance measures.

BROAD INTERPRETATION OF THE GUIDELINES

This kind of error is manifested in inconsistent and/or missing markup, as illustrated, for example, in the following table (Fig. 6), which shows counts of different markup patterns either for relatively frequent temporal expressions (such as the first three entries), or for very similar ones (the last three).

text	time	date	duration	signal	none
"currently"	2	8			4
"recently"	2	10		1	4
"already"	1	1		13	17
"two-week-old"			*		
"[8-month]-old"			*		
"136-years-old"					*

Fig.6: Inconsistent/missing markup.

A different kind of inconsistency, also indicative of less than rigorous application of the guidelines is reflected in the fluidity of placement of left boundary to TIMEX3 expressions in particular. Determiners, pre-determiners and the like tend to float in and out of annotations. In different contexts, TimeBank including the determiner in its span. Similarly, `"<timex3>the late 1970s</timex3>"` and `"the <timex3>late 1950s</timex3>"` are tagged as time expressions which do, or do not, consume the determiner; a behaviour repeatedly observed in the corpus: consider `"the <timex3> early years</timex3>"` vs. `"<timex3>the early 1980s</timex3>"` or `"<timex3>the early summer</timex3>"`.

Clearly, once we become aware of this kind of error, it is possible to make some provisions to accommodate it (thus we define a 'lenient' regime for admitting TIMEX3's, for the purposes of evaluating against TimeBank, (Boguraev and Ando, 2005a)). However, this phenomenon is not limited to time expressions alone, nor can it be counteracted in isolation. For instance, consider the TimeBank analyses of `"<timex3>later this afternoon</timex3>"` and `"<signal>later</signal> <timex3>this month</timex3>"`. Interference is now spread to a different TimeML component analysis; and, arguably, without a SIGNAL in the stream, a subsequent TLINK derivation might be compromised—a situation further exemplified by yet more examples of inconsistent analyses in the corpus:

- `"at <timex3>this crucial moment</timex3>"` vs. `"<timex3>at the moment</timex3>"` and `"<signal>at</signal> the <timex3>end of November</timex3>"`,
- `"<signal>at</signal> <timex3>the beginning of October</timex3>"` and `"<signal>at</signal> the end of October"`.

These are not isolated errors. Fig. 7 shows a subset of a 48-strong list of TIMEX3 expression, typed as TIME.

value in TIMEBANK	covered text
1991-02-24	*yesterday*
1991-02-25	*weekend*
1990-08	*ast August*
1991-02-25	*next few days.*
1988	*last year*
1989-11	*end of November*
1989-Q3	*third-quarter*
1988-Q3	*the year-ago quarter*
1989-03	*March*
1988-Q3	*A year earlier*
1989	*Earlier this year*
1990-Q1	*early 1990*
1989-10-01	*earlier this year*
1989	*now*
1989-10	*this month*

Fig.7: TimeBank markup of TIME expressions, with values incompatible with TIME normalisation guidelines.

The list was derived by a simple projection, against the TimeBank corpus, of searching for TIMEs which might have internal inconsistencies between their TIMEX3 types and values. Syntactically, at least, these TIME expressions are in conflict with the annotation guidelines: for instance, most of their value attributes do not contain the qualifier `"T"` (strongly, if not mandatorily, expected in TIME values); some of them explicitly contain a granularity marker `"Q"` (for year-quarter), which does not conform to the definition of TIME that "the expression [should] refer to time of the day, even if in a very indefinite way", (Saurí et al., 2005):p. 22); and so forth.

To put this projection further into perspective, there are 63 TIME expressions in the corpus (not counting TIMEs in metadata): 48 suspect entries constitute approximately three quarters of the set.

ERRORS IN EVENT AND TLINK MARKUP

As we observed in Section 2, the event typing task is inherently complex. TimeBank exhibits a variety of error in marking EVENTs. Some are more systematic than others: for instance, there is pervasive confusion between money amounts and occurrence events. Some may be due to oversight (or fatigue): a number of verbs are not marked as EVENTs, even if they clearly denote eventualities; the same verb (*"run"*, *"fall"*)—in similar contexts—is marked either as an occurrence or an i_action.

TLINK typing is equally (if not even more so) complex, and we attributed to the difficulties of this task the relatively low performance of our TLINK type classifier (Section 2, and (Boguraev and Ando, 2005b)).

- `In <timex3> the nine months </timex3>, net income <event> rose </event> 4.3% to $525.8 ...`
  `<tlink type=is_included ... />`
- `... said that its net income <event> rose </event> 51% in <timex> the third quarter </timex>`
  `<tlink type=during ... />`

Fig.8: Different TLINK type assignment; similar contexts.

The guidelines (and common sense analysis) suggest that is_included type should be assigned if the time point or duration of EVENT *is included* in the duration of the associated TIMEX3. during, on the other hand, should be assigned as a type if some relation represented by the EVENT *holds during* the duration of the TIMEX3. We note that for this particular typing problem, the subtle distinctions are hard even for human annotators: the TimeBank corpus displays a number of occasions where inconsistent tagging is evident, as Fig. 8 illustrates.

5 CONCLUSION

As we have argued elsewhere (Boguraev and Ando, 2005b), there is some recourse to the problem of paucity of training data. Our studies here, however, show that with a very

small corpus, the 'knock-on' effects of noise are considerably more impactful. The quantitative analysis of the TimeBank corpus in Section 3 offers some indication of a desired size for a training resource for a task with the complexity of TimeML annotation.

The message from our qualitative studies of the corpus is different. As we are primarily motivated by a desire to understand how to interpret the performance figures characteristic of our TimeML annotator (ibid.), our analysis is more focused than just cataloguing errors of omission/errors of commission. Instead, we offer a more detailed breakdown of error types, distinguishing among errors caused by certain features of the annotation-making infrastructure, errors traceable to broader (than intended) interpretation of the guidelines, errors due to the inherent complexity of the task (even as human annotators are concerned), and errors affecting different TimeML components in different—but systematic—ways.

Our position is that understanding the range of categories of error in the corpus makes for informed decisions with regard to how to improve performance of certain TimeML analysis sub-tasks: for instance, detecting certain errors in egregiously wrong annotation (in the corpus) suggests that offending documents might be removed from the training data altogether; observing inconsistencies with a particular subtype of temporal expression might license the use of supplemental data, outside of TimeBank but still compatible with the task; and confirming that a certain TimeML component presents difficulties for consistent detection even to a human annotator is a strong indicator that ultimately, a larger and more consistent TimeBank is crucially required for high quality TimeML analysis.

This is, in fact, the overall conclusion and message from our studies. They are by no means to be taken as a criticism of the corpus, which, as already discussed, was never designed to be a proper training dataset. It is clear that even a relatively minor effort of cleaning up the existing data would improve the overall corpus quality. Such cleanup operation would largely focus on fixing both the errors of omission and commission in the original TimeBank.

Our argument, however, goes further than this: for reasoning engines to function, TimeML analysers need to be built. This speaks to the need to build a training corpus which is larger, broader, and subject to the rigorous processes and controls such as double annotation and inter-annotator agreement, which are by now part of the established methodology of linguistic resource instantiation.[8]

In such a context, our analysis of the TimeBank corpus strongly motivates the need for such an effort, especially in the light of the encouraging performance results of the TimeML parsing machinery we have developed on the basis of TimeBank as it stands.

6 BIBLIOGRAPHICAL REFERENCES

B. Boguraev and R. K. Ando. 2005a. TimeBank-driven TimeML analysis. In J. Pustejovsky, G. Katz and F. Schilder, editors, *International Workshop on Annotating, Extracting, and Reasoning with Time*, Dagstuhl, Germany; <http://www.dagstuhl.de/05151/> [date of citation: 2006-02-16].

B. Boguraev and R. K. Ando. 2005b. TimeML-compliant text analysis for temporal reasoning. In *Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, Edinburgh, Scotland.

R. Fikes, J. Jenkins, and G. Frank. 2003. JTP: A system architecture and component library for hybrid reasoning. Technical Report KSL-03-01, Knowledge Systems Laboratory, Stanford University.

B. Han and A. Lavie. 2004. A framework for resolution of time in natural language. *TALIP Special Issue on Spatial and Temporal Information Processing*, 3(1):11–35.

J. Hobbs and F. Pan. 2004. An ontology of time for the semantic web. *TALIP Special Issue on Spatial and Temporal Information Processing*, 3(1):66–85.

J. Hobbs and J. Pustejovsky. 2004. Annotating and reasoning about time and events. In *AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, Stanford, CA, March.

J. Pustejovsky, J. Castaño, R. Ingria, R. Saurí, R. Gaizauskas, A. Setzer, G. Katz, and D. Radev. 2003a. TimeML: Robust specification of event and temporal expressions in text. In *AAAI Spring Symposium on New Directions in Question-Answering (Working Papers)*, pages 28–34, Stanford, CA.

J. Pustejovsky, P. Hanks, R. Saurí, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, and M. Lazo. 2003b. The TIMEBANK corpus. In Tony McEnery, editor, *Corpus Linguistics*, pages 647–656, Lancaster.

J. Pustejovsky, I. Mani, L. Bélanger, B. Boguraev, B. Knippen, J. Littman, A. Rumshisky, A. See, S. Symonenko, J. Van Guilder, L. Van Guilder, M. Verhagen, and R. Ingria. 2003c. Graphical annotation kit for TIMEML. Technical report, TANGO (TIMEML Annotation Graphical Organizer) Workshop. Version 1.4, <http://www.timeml.org/tango> [date of citation: 2005-06-20].

R. Saurí, J. Littman, B. Knippen, R. Gaizauskas, A. Setzer, and J. Pustejovsky. 2005. TimeML annotation guidelines. Technical report, TERQAS Workshop. Version 1.4, <http://timeml.org/site/publications/timeMLdocs/AnnGuide_1.2.1.pdf> [date of citation: 2006-02-16].

M. Verhagen, I. Mani, R. Sauri, J. Littman, R. Knippen, S. Bae Jang, A. Rumshisky, J. Phillips, and J. Pustejovsky. 2005. Automating temporal annotation with TARSQI. In *43rd Annual Meeting of the Association for Computational Linguistics (ACL-05)*, Ann Arbor, Michigan. Poster/Demo.

[8] Our experiments, and corpus study, have been using TimeBank Version 1.1 (available from http://timeml.org/site/timebank/download.html). The study motivated a revision/ clean-up of the corpus, resulting in Version 1.2; this will be released in early 2006, through the offices of the Linguistic Data Consortium.