TimeBank-Driven TimeML Analysis

Branimir Boguraev and Rie Kubota Ando

IBM T.J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532, USA

bran@us.ibm.com, rie1@us.ibm.com

Abstract. The design of TimeML as an expressive language for temporal information brings promises, and challenges; in particular, its representational properties raise the bar for traditional information extraction methods applied to the task of text-to-TimeML analysis. A reference corpus, such as TimeBank, is an invaluable asset in this situation; however, certain characteristics of TimeBank—size and consistency, primarily—present challenges of their own. We discuss the design, implementation, and performance of an automatic TimeML-compliant annotator, trained on TimeBank, and 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. The results we report are encouraging in the light of a close analysis of TimeBank; at the same time they are indicative of the need for more infrastructure work, especially in the direction of creating a larger and more robust reference corpus. [1]

Keywords. information extraction, machine learning, corpus analysis.

1 Introduction

Developed as a 'transport mechanism' for temporal information, connecting processes of its extraction from a text document followed by a formalisation by means of an ontology of time [1], TimeML [2] 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).[2]

Such representational principles adhere to established guidelines for text markup, in line with prevalent methodology of creating community-wide annotated resources, where linguistic analysis is captured by means of a range of tags with suitably defined attributes for finer-grained specification of analytical detail. TimeML takes these ideas to an extreme, developing half-a-dozen entity and relation marking tags—both consuming and non-consuming—and

[1] This work was supported in part by the ARDA NIMD (Novel Intelligence and Massive Data) program PNWD-SW-6059.

[2] We assume in this paper some familiarity with TimeML. For details of the markup language for time, readers are referred to [3].

Dagstuhl Seminar Proceedings 05151
Annotating, Extracting and Reasoning about Time and Events
http://drops.dagstuhl.de/opus/volltexte/2005/335

defining a large number of attributes for most of them. Consequently, the resulting language is both very expressive (a necessity, arising from the richness of time information and depth of temporal analysis, and addressed from the beginning of the design effort) and very complex (at least in comparison with markup schemes for "named entity" foci of traditional information extraction endeavours).

This is not surprising. Most markup schemes for IE to date target relatively simple phenomena; unlike TimeML, their design has not been informed by the need to capture the variety and complexity of information required to support reasoning.[3] TimeML, in contrast, aims to capture all of the temporal characteristics in a text document, so that the intricate temporal linking among all time expressions and events can then get fully mapped onto an ontologically-grounded temporal graph (or its equivalent), cf. [6], [7]. Indeed, such a mapping (see [8] for a sketch) has been one of the guiding principles in the conception and design of TimeML.

A particularly relevant question, then, concerns the extent to which TimeML-compliant analysis can be automated: temporal reasoning frameworks crucially require such analysis for any practical understanding of time: "... the [TimeML] annotation scheme itself, due to its closer tie to surface texts, can be used as the first pass in the syntax-semantics interface of a temporal resolution framework such as ours. The more complex representation, suitable for more sophisticated reasoning, can then be obtained by translating from the annotations." [7].

Analysis into TimeML is the larger question addressed by this paper. Furthermore, our investigation starts from a particular premise, shared by the designers of TimeML, and in line with the prevalent methodology of developing analytical frameworks on the basis of common, standard, annotated resources.

The TERQAS effort (Temporal and Event Recognition for QA Systems; `http://www.timeml.org/terqas/index.html`), which over the last 24 months coordinated a series of definitional and follow-up workshops from which emerged the current set of TimeML annotation guidelines, also produced a corpus annotated by following those guidelines. A description of the TimeBank corpus [9] states:

> "TIMEBANK contains [186] newswire articles with careful, detailed annotations of terms denoting events, temporal expressions, and temporal signals, and, most importantly, of links between them denoting temporal relations. This collection, the largest temporal-event annotated corpus to date, provides a solid empirical basis for future research into the way texts actually express and connect series of events. It will support research into areas as diverse as the semantics of tense and aspect, the explicit versus implicit communication of temporal relational information, and the variation in typical event structure across narrative domains..."

[3] Arguably, an exception can be found in the 'spirit' of the MUC *event scenario* tasks, which instantiate semantic networks [4, 5]; however, the mapping of an entire text document to a single template can hardly be regarded as logically complete and coherent, in the sense required and assumed by formal event and / or time ontologies.

Additionally, the corpus was regarded as a resource for "training and evaluating algorithms which determine event ordering and time-stamping" (ibid.), as well as providing general-purpose training data for any and all TimeML components. The specific question we ask in this work, then, concerns the extent to which TimeBank does, in fact, support the development of a TimeML-compliant text annotator. As we will see, there are certain characteristics of the corpus—primarily to do with size and consistency—which pose challenges to the notion of using TimeBank as a training resource.

The remainder of this paper presents some analysis of the TimeBank corpus from the point of view of a TimeML annotation task, followed by a description and an evaluation of a hybrid analytical strategy, aiming to make maximal use of the information in the corpus. We argue that tiny as it is (compared to guidelines implicitly established by other information extraction tasks relying on annotated data), TimeBank is still the valuable resource that [9] describes. By developing a strategy for time analysis of text specifically informed by the characteristics of TimeBank—a synergistic approach deploying both finite-state (FS) grammars with broad range of analysis and machine learning techniques capable of also leveraging unannotated data—we demonstrate not only the utility of the corpus as it stands, but more importantly the need for a concerted and concrete effort to create a sizable TimeBank for the use by the larger NLP community.

2 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,

[4] See http://www.itl.nist.gov/iad/894.02/related_projects/muc/main.html.
[5] See http://www.nist.gov/speech/tests/ace/index.htm.

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.

Just how small TimeBank is 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 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 (Temporal Expression Recognition and Normalisation)[8]. TERN only focuses on TIMEX2 (TIMEX3, which extends the TIMEX2 tag [3], is just one of half-a-dozen TimeML components); even so, the TERN training set is almost 800 documents / 300K words-strong.

TimeML tags	# occurrences: 29049	
	marking	non-marking
Timex3	1422	
Signal	2117	
Event	7962	
Instance		7966
ALink		282
SLink		2619
TLink		6681
Total:	11501	17548

Fig. 1: Distributions of TimeML components in TimeBank.

Fig. 1 shows a breakdown of the individual TimeML component distributions in the corpus. While initially the figure of 29K counts of temporally-related entities seems to hold some promise, the perception quickly shifts as we realise that there are only 11.5K marking (text-consuming) spans. Furthermore, given the relationship between EVENTs and INSTANCEs, the 8K INSTANCE tags in the corpus contribute almost nothing to the training cycle (this is particularly true, considering that non-trivial EVENT to event INSTANCE mapping comes

[6] See http://www.cis.upenn.edu/~treebank/home.html.
[7] See http://cnts.uia.ac.be/conll2003/ner/.
[8] See http://timex2.mitre.org/tern.html.

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.

into play in the analysis of time frequencies (SETs), and that only 7 TIMEX3 annotations in the corpus are typed as SETs.)

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.

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

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. 1, 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; see Section 6.3.)

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 (as we shall see in Section 5.2, 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. This is, presumably, a consequence of the relatively eclectic way of collecting TimeBank data, resulting in a less than balanced corpus.

3 Challenges for TimeML analysis

It is clear that temporal annotation is a very complex problem: TimeML was developed precisely to address the issues of complexity and to provide a representational framework capable of capturing the richness of analysis required. One consequence of this is the pervasiveness of relational data which is integral to the underlying representation: all links are, notationally, relations connecting events with other events or temporal expressions. As recent work in relation finding information extraction shows (in particular, in the context of the ACE program), the task requires both some linguistic analysis of text and the definition of complex learning models, typically going beyond just token sequences.

Additionally, as the previous section shows, a different degree of complexity is introduced by the size (and coverage) characteristics of TimeBank. While it may be reasonable to take a position that in our investigation we will focus on those TimeML components which are relatively more prevalent in the data (*e.g.* TLINKs over ALINKs and SLINKs), we still need to address the problem of insufficient training data. Our position thus is that in addition to deploying sophisticated feature generators, we crucially need to leverage machine learning technology capable of exploiting unlabeled data.

Our strategy for TimeML analysis of text develops a hybrid approach utilising both finite-state (FS) grammars over linguistic annotations and machine learning (ML) techniques incorporating a novel learning strategy from large volumes of unlabeled data. The respective strengths of these technologies are well suited for the challenges of the task: complexity of analysis, need for some syntactic and discourse processing, and relative paucity of examples of TimeML-style annotation.

The initial targets of our analysis are TIMEX3 (with attributes), EVENT (plus type), and TLINK (plus type, and limited to links between events and time ex-

pressions); see Figs. 2 and 3. This kind of limitation is imposed largely by the distributional properties of TimeML components annotated in TimeBank (as discussed in Section 2 earlier); but it is also motivated by the observation that to be practically useful to a reasoner, a time analysis framework would need to support, minimally, time stamping and temporal ordering of events. As this is work in progress, the description below offers more details specifically on identifying TIMEX3 expressions, marking and typing EVENTs, and associating (some of these) with TIMEX3 tags (typing the links, as appropriate).

All of these subtasks have components which can be naturally aligned with one or the other of our strategic toolkits. Thus TIMEX3 expressions are intrinsically amenable to FS description, and a grammar-based approach is well-suited to interfacing to the task of TIMEX3 normalisation (*i.e.* instantiating its value). On the other hand, certain attributes of a TIMEX3 (such as temporalFunction, valueFromFunction, functionInDocument) can be assigned by a machine learning component. FS devices can also encode some larger context for time analysis (temporal connectives for marking putative events, clause boundaries for scoping possible event-time pairs, *etc*; see Section 4). To complement such analysis, an ML approach can, using suitable classification methods, cast the problem of marking (and typing) EVENTs as chunking (Section 5.2). As we will see later, a TLINK classifier crucially relies on features derived from the configurational characteristics of a syntactic parse; a result in line with recent work which shows that mid-to-high-level syntactic parsing—typically derived by FS cascades—can produce rich features for classifiers.

In summary, we address the challenges of the TimeBank corpus by combining FS grammars for temporal expressions, embedded in a shallow parser adapted for time analysis, with machine learning trained with TimeBank and unannotated corpora.

4 Finite state devices for temporal analysis

Temporal expressions conform to a set of regular patterns, amenable to grammar-based description. Viewing TIMEX3 analysis as an information extraction task, a cascade of finite-state grammars with broad coverage (compiled down to a single TIMEX3 automaton with 500 states and over 16000 transitions) targets abstract temporal entities such as unit, point, period, relation, *etc*; typically, these will be further decomposed and typed into *e.g.* month, day, year (for a unit); or interval or duration (for a period).

Fine-grained analysis of temporal expressions, instantiating local[9] attributes like granularity, cardinality, ref_direction, and so forth, is crucially required for normalising a TIMEX3: representing "*the last five years*" as illustrated in Fig. 4 below greatly facilitates the derivation of a value (in this case "5PY") for the TIMEX3 value attribute.

[9] "Local" attributes encapsulate time expression characteristics, intrinsic to their temporal nature, but not related to TIMEX3 attributes.

```
[timex : [relative      : true ]
         [ref_direction : past ]
         [cardinality   : 5 ]
         [granularity   : year ] ]
```

Fig. 4: Analysis of a time expression in terms of local attributes.

In effect, such analysis amounts to a parse tree under the TIMEX3. (Not shown above is additional information, anchoring the expression into the larger discourse and informing other normalisation processes which emit the full complement of TIMEX3 attributes—type, temporalFunction, anchorTimeID, *etc*).

It is important to separate the processes of recognition of the span of a TIMEX3 expression from local attribute instantiation for that expression. There is nothing intrinsic to the recognition which necessitates a grammar-based description in preference to a statistical model (as the TERN evaluation exercise demonstrated [10]). However, local attributes (as exemplified above) are necessary for the interpretation rules deriving TIMEX3 value.

TimeBank does not contain such fine-grained mark-up: the grammars thus perform an additional 'discovery' task, for which no training data currently exists, but which is essential for discourse-level post-processing, handling *e.g.* ambiguous and/or underspecified time expressions or the relationship between document-internal and document-external temporal properties (such as 'document creation time').

In addition to *parsing of temporal expressions*, FS devices are deployed for *shallow parsing for feature generation*. We build upon prior work [11], which showed how substantial discourse processing can be carried out from a shallow syntactic base, and derived by means of FS cascading.

Our grammars interleave syntactic analysis with named entity extraction. In particular, they define temporal expressions—as well as other TimeML components, namely events and signals—in terms of *linguistic* units, as opposed to simply lexical cues (as many temporal taggers to date do). The focus on linguistic description cannot be over-emphasised. One of the complex problems for TimeML analysis is that of event identification. A temporal tagger, if narrowly focused on time expressions only (cf. [12]), offers no clues to what events are there in the text. In contrast, a temporal parser aware of the syntax of a time phrase like *"during the long and ultimately unsuccessful war in Afghanistan"* is very close to knowing—from the configurational properties of a prepositional phrase—that the nominal argument (*"war"*) of the temporal preposition (*"during"*) is (most likely) an event nominal.

This kind of information is easily captured within a parsing framework. Additionally, given that EVENTs and LINKs are ultimately posted by a machine learning component, the parser need not commit to *e.g.* event identification and typing. It can gather clues, and formulate hypotheses; and it can then make these available to an appropriate classifier, from whose point of view an EVENT annotation is just another feature. Indeed, the only use of syntactic analysis be-

yond the TIMEX3 parser is to populate a feature space for the classifiers tasked with finding EVENTs and LINKs (Section 5).

Feature generation typically relies on a mix of lexical properties and some configurational syntactic information (depending on the complexity of the task). The scheme we use (Section 5) requires additionally some semantic typing, knowledge of boundaries of longer syntactic units (typically a variety of clauses), and some grammatical function. Fig. 5 illustrates the nature of the FS cascade output.

```
[Snt [svoClause
  [tAdjunct In [NP [timex3 the 1988 period timex3] NP] tAdjunct],
  [SUB [NP the company NP] SUB]
  [VG [GrmEventOccurrence earned grmEventOccurrence] VG]
  [OBJ [NP [Money $20.6 million Money] NP] OBJ] svoClause] ... Snt]
```

Fig. 5: Shallow syntactic analysis (simplified) from finite-state parsing.

Most of the above is self-explanatory, but we emphasise a few key points. The analysis captures the mix of syntactic chunks, semantic categories, and TimeML components used for feature generation (a label like GrmEventOccurrence denotes a hypothesis, generated by the syntactic grammars, that it "earned" is an occurrence type EVENT). It maintains local TIMEX3 analysis; the time expression is inside of a larger clause boundary, with internal grammatical function identification for some of the event predicates. The specifics of mapping configurational information into feature vectors is described in Section 5.

TimeML parsing is thus a bifurcated process of TimeML components recognition: TIMEX3's are marked by FS grammars; SIGNALs, EVENTs and LINKs are putatively marked by the grammars, but the final authority on their identification are classification models built from analysis of both TimeBank and large unannotated corpora. *Features* for these models are derived, as we shall see below, from common strategies for exploiting local context, as well as from mining the results—both mark-up and configurational—of the FS grammar cascading.

5 Classification models for temporal analysis

The classification framework we adopt for this work is based on a principle of *empirical risk minimization*. In particular, we use a *linear classifier*, which makes classification decisions by thresholding inner products of feature vectors and weight vectors. It learns weight vectors by minimizing classification errors (*empirical risk*) on annotated training data.

There are good reasons to use linear classifiers; an especially good one is that they allow for easy experimentation with various types of features, without making any model assumptions. This is particularly important in an investigation like ours, where we do not know *a priori* what kinds of features and feature sets would turn out to be most productive.

For our experiments (Section 6), we use the *Robust Risk Minimization* (RRM) classifier [13], a linear classifier, which has independently been shown useful for a number of text analysis tasks such as syntactic chunking [13], named entity chunking [14–16], and part-of-speech tagging [17].

In marked contrast to generative models, where assumptions about features are tightly coupled with algorithms, RRM—as is the case with discriminative analysis—enjoys clear separation of feature representation from the underlying algorithms for training and classification. This facilitates experimentation with different feature representations, since the separation between these and the algorithms which manipulate them does not require that the algorithms change. We show how choice of features affects performance in Section 6.

To use classifiers, one needs to design *feature vector representation* for the objects to be classified. This entails selection of some predictive attributes of the objects (in effect promoting these to the status of *features*) and definition of mappings between vector dimensions and those attributes (*feature mapping*). Before we describe (later in this section) the essence of our feature design for EVENT and TLINK recognition,[10] we briefly outline word profiling as the enabling technique for counteracting the paucity of training data in TimeBank.

5.1 Word profiling for exploitation of unannotated corpora

In general, classification learning requires substantial amount of labeled data for training—considerably more than what TimeBank offers (Section 2). This characteristic of size is potentially a limiting factor in supervised machine learning approaches. We, however, seek to improve performance by exploiting unannotated corpora, which have the natural advantages of being sizable, and freely available. We use a *word profiling* technique, developed specially for the purposes of exploiting a large unannotated corpus for tagging / chunking tasks [17]. Word profiling identifies, extracts, and manipulates information that characterizes words from unannotated corpora; it does this, in essence, by collecting and compressing feature frequencies from the corpus, a process which maps the commonly used feature vectors to frequency-encoded context vectors.

More precisely, word profiling turns co-occurrence counts of words and features (*e.g.* 'next word', 'head of subject', *etc*) into new feature vectors. For example, observing—in an unannotated corpus—that nouns like *"extinction"* and *"explosion"* are often used as syntactic subject to *"occur"*, and that *"earthquakes"* and *"explosions" "happen"*, helps to predict that *"explosion"*, *"extinction"*, and *"earthquake"* all function like event nominals. In Section 6.2, we demonstrate the effectiveness of word profiling, specifically for EVENT recognition.

[10] We do not discuss SIGNAL recognition here, as the `signal` tag itself contributes nothing to EVENT or TLINK recognition, beyond what is captured by a lexical feature over the temporal connective, independent of whether it is tagged as SIGNAL or not.

5.2 EVENT recognition as a classification problem

Similarly to named entity chunking, we cast the EVENT recognition task as a problem of sequential labeling of tokens by encoding chunk information into token tags. For a given *class*, this generates three tags: E:*class* (the last, end, token of a chunk denoting a mention of *class* type), I:*class* (a token inside of a chunk), and O (any token outside of any target chunk). The example sequence below indicates that the two tokens *"very bad"* are spanned by an event-state annotation.

$$\cdots \textit{another}/\mathsf{O}\ \textit{very}/\mathsf{I{:}event\text{-}state}\ \textit{bad}/\mathsf{E{:}event\text{-}state}\ \textit{week}/\mathsf{O}\ \cdots$$

In this way, the EVENT chunking task becomes a $(2k+1)$-way classification of tokens where k is the number of EVENT types; this is followed by a Viterbi-style decoding. (We use the same encoding scheme for SIGNAL recognition.)

The feature representation used for EVENT extraction experiments mimics the one developed for a comparative study of entity recognition with word profiling [17]. The features we extract are:

- token, capitalisation, part-of-speech (POS) in 3-token window;
- bi-grams of adjacent words in 5-token window;
- words in the same syntactic chunk;
- head words in 3-chunk window;
- word uni- and bi-grams based on subject-verb-object and preposition-noun constructions;
- syntactic chunk types (noun or verb group chunks only);
- token tags in 2-token window to the left;
- tri-grams of POS, capitalisation, and word ending;
- tri-grams of POS, capitalisation, and left tag.

5.3 TLINK recognition as a classification problem

TLINK is a relation between events and time expressions which can link two EVENTs, two TIMEX3's, or an EVENT and a TIMEX3. As we stipulated earlier (Section 3), presently we focus on TLINKs between events and time expressions.

As a relational link, TLINK does not naturally fit the tagging abstraction underlying the chunking problem, as outlined above. Instead, we formulate a classification task as follows. After posting EVENT and TIMEX3 annotations (by the event classifier and the FS temporal parser, respectively), for each pairing between an EVENT and a TIMEX3, we ask whether it is a certain type of TLINK. This defines a $(\ell+1)$-way classification problem, where ℓ is the number of TLINK types (before, after, *etc*). The adjustment term '$+1$' is for the *negative* class, which indicates that the pair does not have any kind of temporal link relation.

The relation-extraction nature of the task of posting TLINKs requires a different feature representation, capable of encoding the syntactic function of the relation arguments (EVENTs and TIMEX3's), and some of the larger context of their mentions. To that end, we consider the following five *partitions* (defined in

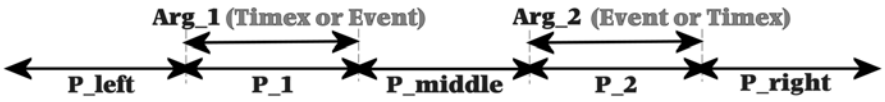

Fig. 6: Partitions for TLINK classifier segmentation.

terms of tokens): spans of arguments (P_1 or P_2); two tokens to the left / right of the left / right argument (P_left / P_right); and the tokens between the arguments (P_middle). From each partition, we extract tokens and parts-of-speech as features (Fig. 6).

We also consider *segments* (*i.e.* syntactic constructions derived by FS analysis: 'when-clause', 'subject', *etc*) in certain relationship to partitions:

- contained in P_1, P_2, or P_middle;
- covering P_1 (or P_2) but not overlapping with P_2 (or P_1);
- occurring to the left of P_1 (or the right of P_2); or
- covering both P_1 and P_2.

We use uni- and bi-grams of types of these segments as features.

In this feature representation, segments play a crucial role by capturing the syntactic functions of EVENTs and TIMEX3's, as well as the syntactic relations between them.

Thus in the example analysis in Fig. 5 (p. 9), `svoClause` is the smallest segment containing both an EVENT and a TIMEX3, which is indicative of (or at least does not prohibit) a direct syntactic relation between the two. In the next example (Fig. 7), the TIMEX3 and EVENT chunks are contained in different clauses (a `thatClause` and a `svoClause`, respectively), which structurally prohibits a TLINK relation between the two. Our feature representation is capable of capturing this information via the types of the segments that contain each of EVENT and TIMEX3 without overlapping.

```
[Snt
 Analysts have complained
 [thatClause that [timex3 third-quarter timex3] corporate earnings
    have n't been very good thatClause]
 [svoClause , but the effect [event hit event] ... svoClause] Snt]
```

Fig. 7: Syntactic configuration discouraging of a TLINK.

6 Experiments

In line with our current investigation focus (as defined in Section 3), we present here performance results on recognition and typing of TIMEX3, EVENT and

TLINK only. Our primary objective here is to report on how effective our analytical strategy is in leveraging the reference nature of the small TimeBank corpus for training classifiers for TimeML. This is the first attempt to build a TimeML-compliant analyser which addresses a more or less full complement of TimeML components; thus there are no comparable results in the literature.

The results (micro-averaged F-measure) reflect experiments with different settings, against the TimeBank corpus, and produced by 5-fold cross validation.

6.1 TIMEX recognition and typing

Fig. 8 presents performance results of our TIMEX3 analysis subsystem. Experiments were carried out under different settings. "Span" refers to strict match of both boundaries (the extent) of a TIMEX3 expression; "sloppy" admits time expressions recognised by the FS grammars as long as their right boundary is the same as the reference expression in TimeBank. (As we will see, in Section 7.2 later, TimeBank is inconsistent with respect to whether some 'left boundary' items—determiners, pre-determiners, and so forth—are marked as a part of the time expression or not; the "sloppy" setting tries to account for this somewhat). As of the time or writing, there are no published results for full TimeML-

Task	P	R	F
Span	77.6	86.1	81.7
Span ('sloppy')	85.2	95.2	89.6
			Accuracy
Type (given 'true' span)			81.5
Span + type	64.5	71.6	67.9
Span ('sloppy') + type	70.1	77.8	73.7

Fig. 8: TIMEX3 analysis results, with / without typing. Typing carried out after / simultaneously with span marking.

compliant analysis. We offer here only indirect assessment of our TIMEX3 analysis task, by observing that the numbers for extent marking are not very far from the best systems performance reported at the TERN conference. Of course, given the different definitions of TIMEX2 and TIMEX3, as well as TimeBank's relatively 'cavalier' attitude with respect to TIMEX3's left boundary, the comparison is not very meaningful; still, it is indicative of some level of grammar coverage, especially given the incommensurate sizes of the TERN training data and the TimeBank corpus (Section 2).

While TIMEX3 spans are determined by grammars, we use a classifier to type the time expressions. Again, this decision was motivated largely by observing

some inconsistencies in type assignment in the corpus, and we felt that, for the purposes of strictly matching the data, machine learning was a more fitting approach to try first (we are yet to compare the typing results presented here with typing by the FS grammars; such a comparison is tied somewhat to getting a better understanding of the quality of annotations in TimeBank; see Section 7 below). The TIMEX3 typing classifier (second segment of Fig. 8) is configured to use "true" TIMEX3 spans, as per TimeBank, as data points, to which it assigns a category (type) label; thus the table gives a single accuracy measure.

Finally, we report on a joint task, which combines (in sequence) extent marking by FS grammars and type determination as classification process over given spans (this classification task, and features, are defined similarly to the IEO scheme used for EVENT extraction and typing, in without-word-profiling setting; see Section 5.2). In effect, the results here confirm the intuition that imperfect subtasks individually contribute to cumulative degradation of performance.

6.2 EVENT recognition and typing

The example analysis in Fig. 5, and the description of features used for the EVENT classification task (Section 5.2) demonstrates how local information and syntactic environment both contribute to the feature generation process. Fig. 9 shows performance results with and without word profiling for exploiting an unannotated corpus. For the word profiling experiments, we extracted feature

features	with typing	w/o typing
basic	61.3	78.6
basic + word-profiling	64.0 (+2.7)	80.3 (+1.7)

Fig. 9: EVENT extraction results, with / without typing. Numbers in parentheses show contribution of word profiling, over using basic features only.

co-occurrence counts from 40 million words of 1991 *Wall Street Journal* articles. The proposed event chunks are counted as correct only when both the chunk boundaries and event types are correct. 64.0% F-measure is lower than typical performance of, for instance, named entity chunking; this result is indicative of the effects of insufficient training data. On the other hand, a strongly positive indicator here is the fact that word profiling clearly improves performance. In a different setting, when we train the EVENT classifiers without typing, we obtain 80.3% F-measure. This confirms the intuition that the EVENT typing task is inherently complex, and requires more training data.

distance (# of TLINKs)	features	with typing	w/o typing
distance $\leq$ 64 tokens (1370 TLINKs)	baseline	21.8	34.9
	basic	52.1	74.1
	basic+FS	53.1 (+1.0)	74.8 (+0.7)
distance $\leq$ 16 tokens (1269 TLINKs)	baseline	38.7	61.3
	basic	52.8	75.8
	basic+FS	54.3 (+1.5)	76.5 (+0.7)
distance $\leq$ 4 tokens (789 TLINKs)	baseline	49.8	76.1
	basic	57.0	80.1
	basic+FS	58.8 (+1.8)	81.8 (+1.7)

Fig. 10: TLINK extraction results, with / without typing. Parentheses show positive contribution of grammar-derived features, over using basic features only. Baseline method posts TLINKs over 'close' pairs of EVENTs and TIMEX3's.

6.3 TLINK recognition and typing

In this experimental setting, we only consider the pairings of EVENT and TIMEX3 which appear within a certain distance in the same sentences (as we will see shortly, this hardly reduces the problem space).[11]

For comparison, we implement the following simple baseline method. Considering the text sequence of EVENTs and TIMEX3's, only 'close' pairs of potential arguments are coupled with TLINKs; EVENT e and TIMEX3 t are close if and only if e is the closest EVENT to t and t is the closest TIMEX3 to e. For all other pairings, no temporal relation is posted. Depending on the 'with-' / 'without-typing' setting, the baseline method either types the TLINK as the most populous class in TimeBank, is_included, or simply marks it as 'it exists'.

Results are shown in Fig. 10. Clearly, the detection of temporal relations between events and time expressions requires more than simply coupling the closest pairs within a sentence (as the baseline does). It is also clear that the baseline method performs poorly, especially for pairings over relatively long distances. For instance, it produces 34.9% (in F-measure) when we consider the pairings within 64 tokens without typing. In the same setting, our method produces 74.8% in F-measure, significantly outperforming the baseline.

We compare performance against two types of feature representation: 'basic' and 'basic+FS grammar', which reflect the without- and with-segment-type information obtained by the grammar analysis, respectively. As the positive delta's show, configurational syntactic information can be exploited beneficially by our process. When we focus on the pairings within a 4-tokens window, we achieve 81.8% F-measure without typing of TLINKs, and 58.8% with typing. (The task without typing is a binary classification to detect whether the pairing

[11] To evaluate the TLINK classifier alone, we use the EVENT and TIMEX3 annotations in TimeBank. Also, note that the focus on links within a sentence span naturally excludes TLINKs with time expressions in document metadata.

has a TLINK relation or not, regardless of the type.) As the figure shows, the task becomes harder when we consider longer distance pairings. Within a 64 token distance, for instance,, we obtain figures of 74.8% and 53.1%, without and with typing respectively.

While we are moderately successful in detecting the *existence* of temporal relations, the noticeable differences in performance between the task settings with and without typing indicate that we are not as successful in distinguishing one type from another. In particular, the major cause of the relatively low performance of TLINK typing is the difficulty in distinguishing between during and is_included link types.

7 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' 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).

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

Fig. 11: Annotation tool gone wrong.

7.1 Annotation infrastructure errors

Consider the (excerpt from an) annotated document illustrated in Fig. 11. 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. Equally problematic are errors likely to be introduced in a pre- (or post-) processing cycle by an XML parser thrown off by malformed XML markup (Fig. 12). The first three examples are, arguably

```
[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. 12: Embedded and overlapping XML annotations.

'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.[12]

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. 11 and Fig. 12 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) [18]. 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.

7.2 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. 13), 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).

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

[12] 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.

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. 13: Inconsistent / missing markup.

to TIMEX3 expressions in particular. We already mentioned that determiners, pre-determiners and the like tend to float in and out of annotations (Section 6.1). In different contexts, TimeBank marks the string *"the fourth quarter"* as a TIMEX3, with or without including the determiner in its span. Similarly, *"[the late 1970s]"* and *"the [late 1950s]"* are tagged as time expressions which do, or do not, consume the determiner; a behaviour repeatedly observed in the corpus: cf. *"the [timex3 early years timex3]"* vs. *"[timex3 the early 1980s timex3]"* or *"[timex3 the early summer timex3]"*.

As we have seen, once we become aware of this kind of error, it is possible to make some provisions to accommodate it (the 'lenient' definition of admitting TIMEX3's, in Section 6.1 above, is an example). 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. 14 shows a subset of a 48-strong list of TIMEX3 expression, typed as TIME. The list was derived by a simple projection, against the TimeBank corpus, of searching for TIMEs which might have internal inconsistencies. 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 DATE 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", [3]:p. 22).

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

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

7.3 Errors in EVENT and TLINK markup

As we observed in Fig. 9 (Section 6.2), 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 6.3).

- *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. 15: Different type assignment to TLINKs from 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. 15 illustrates.

8 Conclusion

The primary focus of this investigation has been the study—direct and indirect—of the characteristics of the TimeBank corpus which are likely to influence its utility as a training resource for developing automatic TimeML analysis machinery. Additionally, we have used the overall task to experiment with a specially developed strategy for leveraging minuscule amounts of training data. The strategy synergistically blends finite-state analysis for shallow syntactic parsing with a machine learning technique. Especially novel components of this blend are the aggressive analysis, by a complex grammar cascade, targeting considerably more than just partitioning text into chunks; coupled with an extension of the learning component, specifically designed to counteract paucity in pre-annotated data with the ability to train over unannotated data as well as exploit whatever labeled data is available, no matter how small.

The performance results, in particular where the novel components of EVENT and TLINK analysis are targeted, appear to fall short of expectations in line with current state-of-the-art information extraction capabilities. However, given the extreme paucity of the available reference data, as well as the inherently noisy nature of a corpus which has not been designed and populated using rigorous methods for generating training data, our experience is indicative of the effectiveness of a hybrid analytical approach. Furthermore, it is clear that with the ability to use unannotated corpora in conjunction with TimeBank, even small improvements to the corpus would significantly boost performance.

Given that ours is the first systematic attempt at TimeML-compliant analysis, aiming at a more or less full complement of TimeML components, there are no comparable results in the literature. [19] discuss some pioneering work in linking events with times, and ordering events, suggestive of productive strategies for posting (some) TLINK information. However, the nature of these efforts is such that differences in premises, representation, and focus make a direct performance comparison impossible. Furthermore, the work pre-dates TimeML, and cannot be conveniently mapped to TimeBank data; this, in effect, precludes a quantitative comparison with our work.

Most recently, the TARSQI project[13] has been developing strategies and heuristics for particular subsets of TimeML components [20]; again, there is no basis for direct comparison, as only partial overlap exists between the phenomena and attributes targeted by that work and ours. For this reason, as well as because TARSQI does not explicitly focus on investigating the utility of TimeBank as a training resource, it is not constructive to attempt comparative assessment.

Our analysis of TimeBank confirms that, from the point of view of developing strong models of temporal phenomena, the corpus would benefit from the

[13] See `http://www.timeML.org/tarsqi/`.

application of rigorous methodology for compiling training data. It is clear—especially from considering the results presented in Section 6 and the corpus characteristics highlighted in Section 7—that even a relatively minor effort of cleaning up the existing data would improve performance. Such cleanup operation would largely focus on fixing both the errors of omission and commission in the original TimeBank.

More challenging, but also more productive and useful to the community, would be an effort to create a larger TimeBank which—by virtue of the systematic methods of developing an annotated corpus within an established set of annotation guidelines—will truly become the widely usable reference resource envisaged, from the outset of the TimeML definition and standardisation effort, by its creators [9].

References

1. Hobbs, J., Pan, F.: An ontology of time for the semantic web. TALIP Special Issue on Spatial and Temporal Information Processing **3** (2004) 66–85
2. Pustejovsky, J., Castaño, J., Ingria, R., Saurí, R., Gaizauskas, R., Setzer, A., Katz, G., Radev, D.: TimeML: Robust specification of event and temporal expressions in text. In: AAAI Spring Symposium on New Directions in Question-Answering (Working Papers), Stanford, CA (2003) 28–34
3. Saurí, R., Littman, J., Knippen, B., Gaizauskas, R., Setzer, A., Pustejovsky, J.: TimeML annotation guidelines. Technical report, TERQAS Workshop (2005) Version 1.4, <http://timeml.org/timeMLdocs/AnnGuide14.pdf> [date of citation: 2005-06-20].
4. Advanced Research Projects Agency: Proceedings of the Sixth Message Understanding Conference (MUC-6), Advanced Research Projects Agency, Software and Intelligent Systems Technology Office (1995)
5. Advanced Research Projects Agency: Proceedings of the Seventh Message Understanding Conference (MUC-7), Advanced Research Projects Agency, Software and Intelligent Systems Technology Office (1998)
6. Fikes, R., Jenkins, J., Frank, G.: JTP: A system architecture and component library for hybrid reasoning. Technical Report KSL-03-01, Knowledge Systems Laboratory, Stanford University (2003)
7. Han, B., Lavie, A.: A framework for resolution of time in natural language. TALIP Special Issue on Spatial and Temporal Information Processing **3** (2004) 11–35
8. Hobbs, J., Pustejovsky, J.: Annotating and reasoning about time and events. In: AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning, Stanford, CA (2004)
9. Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., Lazo, M.: The TIMEBANK corpus. In McEnery, T., ed.: Corpus Linguistics, Lancaster (2003) 647–656
10. DARPA TIDES (Translingual Information Detection, Extraction and Summarization): The TERN evaluation plan; time expression recognition and normalization. Working papers, TERN Evaluation Workshop (2004)
11. Kennedy, C., Boguraev, B.: Anaphora for everyone: Pronominal anaphora resolution without a parser. In: Proceedings of COLING-96 (16th International Conference on Computational Linguistics), Copenhagen, DK (1996)

12. Schilder, F., Habel, C.: Temporal information extraction for temporal QA. In: AAAI Spring Symposium on New Directions in Question-Answering (Working Papers), Stanford, CA (2003) 35–44
13. Zhang, T., Damerau, F., Johnson, D.E.: Text chunking based on a generalization of Winnow. Journal of Machine Learning Research **2** (2002) 615–637
14. Florian, R., Ittycheriah, A., Jing, H., Zhang, T.: Named entity recognition through classifier combination. In: Proceedings of CoNLL-2003. (2003)
15. Zhang, T., Johnson, D.E.: A robust risk minimization based named entity recognition system. In: Proceedings of CoNLL-2003. (2003) 204–207
16. Florian, R., Hassan, H., Jing, H., Kambhatla, N., Luo, X., Nicolov, N., Roukos, S.: A statistical model for multilingual entity detection and tracking. In: Proceedings of HLT-NAACL'04. (2004)
17. Ando, R.K.: Exploiting unannotated corpora for tagging and chunking. In: Proceedings of ACL-04. (2004)
18. Pustejovsky, J., Mani, I., Bélanger, L., Boguraev, B., Knippen, B., Littman, J., Rumshisky, A., See, A., Symonenko, S., Guilder, J.V., Guilder, L.V., Verhagen, M., Ingria, R.: Graphical annotation kit for TIMEML. Technical report, TANGO (TIMEML Annotation Graphical Organizer) Workshop (2003) Version 1.4, <http://www.timeml.org/tango> [date of citation: 2005-06-20].
19. Mani, I., Pustejovsky, J., Sundheim, B.: Introduction: special issue on temporal information processing. ACM Transactions Asian Language Information Processing **3** (2004) 1–10
20. Verhagen, M., Mani, I., Sauri, R., Littman, J., Knippen, R., Jang, S.B., Rumshisky, A., Phillips, J., Pustejovsky, J.: Automating temporal annotation with TARSQI. In: 43rd Annual Meeting of the Association for Computational Linguistics (ACL-05), Ann Arbor, Michigan (2005) Poster / Demo.