Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 2919–2927
Marseille, 20-25 June 2022

A Comprehensive Evaluation and Correction of the TimeBank Corpus

Mustafa Ocal, Antonela Radas, Jared Hummer, Karine Megerdoomian*, & Mark A. Finlayson
Florida International University
Knight Foundation School of Computing and Information Sciences
CASE Building, Room 362, 11200 S.W. 8th Street, Miami, FL USA 33199
MITRE Corporation, 7515 Colshire Dr, McLean, VA USA 22102*
{mocal001, arada002, jhumm001, markaf}@fiu.edu
karine@mitre.org*

Abstract

TimeML is an annotation scheme for capturing temporal information in text. The developers of TimeML built the TimeBank corpus to both validate the scheme and provide a rich dataset of events, temporal expressions, and temporal relationships for training and testing temporal analysis systems. In our own work we have been developing methods aimed at TimeML graphs for detecting (and eventually automatically correcting) temporal inconsistencies, extracting timelines, and assessing temporal indeterminacy. In the course of this investigation we identified numerous previously unrecognized issues in the TimeBank corpus, including multiple violations of TimeML annotation guide rules, incorrectly disconnected temporal graphs, as well as inconsistent, redundant, missing, or otherwise incorrect annotations. We describe our methods for detecting and correcting these problems, which include: (a) automatic guideline checking (109 violations); (b) automatic inconsistency checking (65 inconsistent files); (c) automatic disconnectivity checking (625 incorrect breakpoints); and (d) manual comparison with the output of state-of-the-art automatic annotators to identify missing annotations (317 events, 52 temporal expressions). We provide our code as well as a set of patch files that can be applied to the TimeBank corpus to produce a corrected version for use by other researchers in the field[1].

Keywords: TimeML, TimeBank, Annotation Evaluation, Automatic Annotation Correction

1. Introduction

TimeML is a temporal annotation scheme for capturing events, temporal expressions, and temporal relations in natural language texts (Pustejovsky et al., 2003a). The TimeBank corpus was released as a reference corpus for TimeML which provides extensive annotated data for training and testing temporal analysis systems that produce TimeML (Pustejovsky et al., 2003b). The TimeBank corpus has been used in much Natural Language Processing (NLP) research, including works on event detection (UzZaman and Allen, 2010; Färber and Rettinger, 2015; Bansal et al., 2018; Veyseh et al., 2021), temporal expression recognition (Kolomiyets and Moens, 2009; Zhong et al., 2017; Chen et al., 2019), and temporal link extraction (Mani et al., 2006; Mirroshandel and Ghassem-Sani, 2011; Kadir et al., 2016; Ning et al., 2018). If there are errors in TimeBank, they could potentially affect the accuracy of all works that use it.

In prior work we have been using TimeBank as a starting point for developing methods for detecting temporal inconsistency, measuring temporal indeterminacy, and automatically correcting and enriching temporal graphs. In the course of that work, we have identified a number of previously unrecognized issues in the TimeBank corpus, including numerous violations of TimeML annotation guide rules, incorrectly disconnected temporal graphs, as well as inconsistent, redundant, missing, or otherwise incorrect annotations. Here we describe these issues and present a suite of methods for correcting the corpus. First, we implemented a system to check whether TimeBank follows the TimeML annotation guidelines where they can be formulated as strict, syntactic, no-exceptions rules. This revealed a number of problems. Second, for each annotated file we checked overall temporal consistency, which revealed additional problems not described in prior work. This includes checking for redundant or inconsistent self-loops in the graphs. Third, we analyzed the connectivity of the TimeML graphs, and this analysis as a guide showed that numerous TimeML graphs were improperly disconnected. Finally, we used an automatic TimeML parser (CAEVO) to parse raw TimeBank files, and manually compared those automatic annotations with the gold-standard annotations, thereby identifying a number of places where the TimeBank corpus misses events, times, and relations.

In particular, we show that TimeBank has 109 instances that violate the strict TimeML annotation guidelines rules. We also show that roughly 1/3 of the TimeBank files include an inconsistent cycle in the extracted TimeML graphs, which includes 15 inconsistent self-loops. Additionally, we detected on average 8.1 discon-

[1]Data and code may be downloaded from https://doi.org/10.34703/gzx1-9v95/EFNL6H. Approved for Public Release; Distribution Unlimited. Public Release Case Number 22-0288. ©2022 The MITRE Corporation and Florida International University. ALL RIGHTS RESERVED

nected graphs per file which suggested potentially up to 4 missing temporal relations per file. We also discovered 10 redundant self-loops (a loop that doesn't cause inconsistency but also doesn't provide additional information, and is not signaled in the text). After we manually compared gold-standard annotations and CAEVO-based annotations, we identified 317 missing events, 52 missing temporal expressions, and 369 additional missing TimeML links.

The detected errors of the TimeBank corpus are the result of either incorrect or missing annotations. As we have shown through extensive experimentation elsewhere, errors such as those described above can have dramatic effects on the quality of the final graphs and downstream tasks (Ocal et al., 2022). We corrected the errors by adding or changing the necessary event, temporal expression, or TimeML link. The correction process was a double annotation process. First, the authors took the CAEVO-generated patch file and annotated it separately. Then, the authors got together and went through each annotation one by one. In total, we corrected 317 events, 52 temporal expressions, and 1,265 TimeML links, which correspond to 4%, 4%, and 13% of the total. We released both our analysis code and diff files that allow other researchers in the field to apply our corrections to their own copies of TimeBank.

The paper is organized as follows. First, we review prior work on TimeML, the TimeBank corpus, automatic TimeML annotation, and TimeBank evaluation (§2. Next, we explain our evaluation methods in detail (§3), followed by our results and an explanation of how the corrections were made (§4). Finally, we conclude with a discussion (§5) and provide a summary of contributions (§6).

2. Related Work

2.1. TimeML

TimeML is an SGML-based annotation scheme to annotate temporal information in documents (Saurí et al., 2006). TimeML is built on Allen's interval algebra (Allen, 1983) which allows defining 13 possible temporal relations between events. Using TimeML, we can annotate events and temporal expressions in documents. An event is a situation that happens or occurs. There are seven types of events: REPORTING, PERCEPTION, ASPECTUAL, I_ACTION, I_STATE, STATE, and OCCURRENCE. An example of an OCCURRENCE event is shown below.

(1) She **wrote** the letter.

A temporal expression is any word or phrase that represents a date, a time interval, a time point, or a duration:

(2) He was in Miami on **Tuesday**.

TimeML allows for relationships (links) between events and temporal expressions. There are three types of TimeML links: Temporal Link (TLINK), Aspectual Link (ALINK), and Subordination Link (SLINK).

A TLINK represents a temporal relationship between events and temporal expressions. There are 14 types of TLINKs: BEFORE, AFTER, I_BEFORE, I_AFTER, SIMULTANEOUS, IDENTITY, BEGINS, BEGUN_BY, ENDS, ENDED_BY, INCLUDES, IS_INCLUDED, DURING, and DURING_INV. In the example below, the AFTER TLINK represents a temporal relationship between *drank* and *ate*.

(3) He **drank** tea after he **ate** his breakfast.
(drank $-$AFTER$\rightarrow$ ate)

An ALINK holds a relationship between an aspectual event and its argument event. There are five types of ALINKs: INITIATES, TERMINATES, CULMINATES, REINITIATES, and CONTINUES. In the following example, INITIATES ALINK represents an aspectual relationship between *started* and *watching*.

(4) We **started** **watching** the movie.
(started $-$INITIATES$\rightarrow$ watching)

An SLINK holds a relationship between a subordination event and its argument event. There are six types of SLINKs: MODAL, FACTIVE, COUNTER_FACTIVE, EVIDENTIAL, NEG_EVIDENTIAL, and CONDITIONAL. An example of a MODAL SLINK is shown below.

(5) Kai **promised** to **play** basketball with me.
(promised $-$MODAL$\rightarrow$ play)

Annotating a text with TimeML naturally begets a TimeML graph, which is a graph where nodes are events and temporal expressions, and edges are TimeML links. An illustration of a TimeML graph (drawn from *wsj_1073.tml*) is shown in Figure 1.

(6) [**DCT:10/25/1989**$_1$]: Advanced Medical Technologies Inc. **said**$_2$ it **purchased**$_3$ 93% of a unit of Henley Group Inc. Advanced Medical **paid**$_4$ $106 million in cash for its share in a unit of Henley's Fisher Scientific subsidiary. The unit makes intravenous pumps used by hospitals and **had**$_5$ more than $110 million in **sales**$_6$ **last year**$_7$, according to Advanced Medical.

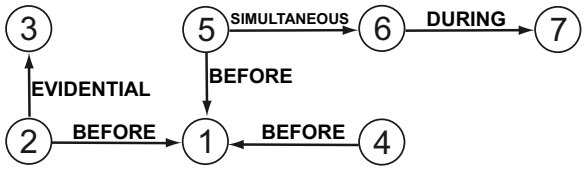

Figure 1: Visualization of the gold-standard TimeBank TimeML graph for Example (6).

2.2. TimeBank

The TimeBank corpus was developed as a reference corpus for the TimeML annotation scheme (Pustejovsky et al., 2003b). More than 50 researchers worked on it including professors, graduate students, and undergraduate students, in both linguistics and computer science. It comprises news stories from various sources such as ABC, CNN, and The Wall Street Journal. TimeBank 1.1 was the first release, which suffered from problems due to the flaws in the annotation software, as well as internal inconsistency such as different values for the same Timex (temporal expression) trigger words or having no values for some TimeML objects. It was substantially revised for TimeBank 1.2, which is the version we use here.

The TimeBank corpus contains a rich set of TimeML annotations. It comprises 183 texts in which are annotated 7,935 events, 1,414 temporal expressions, 6,418 TLINKs, 265 ALINKs, and 2,932 SLINKs, and it has been used for many NLP tasks such as question answering (UzZaman et al., 2012a), event detection (Färber and Rettinger, 2015), temporal expression recognition (Zhong et al., 2017), temporal representation (Ocal and Finlayson, 2020), temporal link extraction (Mirza and Tonelli, 2016) and temporal annotation evaluation (UzZaman et al., 2012b).

2.3. Automatic TimeML Annotators

TimeBank enabled many researchers in the field to develop automatic TimeML annotators. The TARSQI tool (Verhagen et al., 2005) comprises five modules to annotate temporal information in documents. TARSQI takes a raw text as input, and first, it recognizes temporal expression with the GUTime module. Second, it detects events with Evita. Then, it detects TimeML links with GUTenLINK and SLinknet. Finally, it performs temporal reasoning with the SputLink module and produces TimeML annotated text.

ClearTK (Bethard, 2013) is a pipeline of three supervised models that use only a small set of features. The time module uses time related features such as text, stem, POS-tags, and the temporal type of each alphanumeric sub-token derived from time words. The event module uses aspect, modality, tense, POS tags, and event attributes. And finally, the temporal relation module uses event and time features to predict TLINKs between events and times.

CAEVO (Chambers et al., 2014) is a sieve-based architecture that contains four modules to annotate documents: (1) SUTime (Chang and Manning, 2013) is a rule based system that recognizes and normalizes time expressions using text regex, compositional rules, and filtering rules. (2) NavyTime (Chambers, 2013) is a maximum entropy classification framework based on the lexical and syntactic features such as n-grams, POS tags, lemmas, typed dependencies and WordNet events. (3) The temporal relation extraction module is a supervised machine learning module to extract relations between events and times in the same sentences and neighboring sentences. (4) The dense ordering module applies transitive closure to extract dense TimeML annotations. When using automatic annotations to drive manual correction of TimeBank, as described later in the paper, we choose CAEVO because CAEVO has the best performing independent models in the form of SUTime (0.92 F_1), NavyTime (0.81 F_1), and CAEVO-TLINK (0.51 F_1).

2.4. Prior TimeBank Evaluations

Several previous works have provided different types of analysis for the TimeBank corpus.

Boguraev and Ando (2006) evaluated the first version of the TimeBank corpus (TimeBank 1.1). They presented a quantitative analysis of the TimeBank corpus such as distribution of relations, event classes, Timex types, and TimeML components. They showed that the annotation tool used to construct TimeBank caused a systematic shift by a single character. They also showed that for the same Timex signal, TimeBank 1.1 had different types of (or missing) Timex tags.

Similarly, Boguraev et al. (2007) presented a quantitative analysis not only for TimeBank 1.1 but also TimeBank 1.2, which allows them to compare two corpora. They selected a random document from the corpora and evaluated it manually to compare the number of errors between TimeBank 1.1 and TimeBank 1.2. Based on their results, the chosen document contained 96 errors (8 timex, 32 event, 43 links, and 13 signals) for TimeBank 1.1 and 28 errors (1 timex, 10 events, and 17 links) for TimeBank 1.2, suggesting that TimeBank 1.2 was indeed an improvement over the prior version.

Caselli and Morante (2018) presented a detailed error analysis for automatic temporal processing systems that were submitted to TempEval-3. They manually evaluated 15% of the TimeBank corpus to check why automatic temporal processing systems failed to detect temporal relations in the corpus. The results showed that plenty of gold temporal relations are either wrong or in dispute, with the resulting suggestion that annotators consider event's tense and aspect while annotating gold temporal relations.

Inel and Aroyo (2019) compared the TimeBank corpus with other TimeML annotated corpora by manually evaluating events in each sentence. The results showed that TimeBank contains sentences that do not have any events, and there are a number of events that are not consistent with annotation guidelines. The results also showed that in some cases, the same phrases are tagged differently in corpora. For example, "election day" was annotated as TIMEX3 in TimeBank while in other corpora "election" was labeled as an event. Finally, the comparison showed that the TimeBank corpus has only a single token for events while other corpora have multi-token events as well.

Ocal and Finlayson (2020) extracted timelines from the TimeBank corpus and presented a quantitative analysis

on the timelines as well as the evaluation of the temporal indeterminacy of the timelines. They reported the timelines extracted from TimeBank have an average of 9.3 time steps and 51.1 time points. Additionally, the timelines have a 67.9% indeterminacy score.

The Corpus Analysis and Validation for TimeML (CAVaT) tool (Derczynski and Gaizauskas, 2012) is the most similar work to that presented here. CAVaT is a sanity check system for TimeML annotated corpora which checks the temporal consistency of TLINKs, identifies disconnected subgraphs (TLINKs only), and detects self-loops. Additionally, CAVaT prints out the TLINK distribution and shows how many TLINKs are triggered by temporal signals. CAVaT was run over the TimeBank corpus and detected 30 inconsistent texts, 26 self-loops, and showed that no text has a fully connected TLINK graph. In our work, we go further than CAVaT by checking not only the consistency of TLINKs, but of the entire TimeML graph. In contrast to CAVaT we check the disconnectivity of the entire TimeML graphs (again, not just TLINKs), and moreover provide automatic corrections for them.

3. Methods

3.1. Strict TimeML Annotation Rules

The first category of corrections we examine are rules from the TimeML annotation guide that are strictly syntactic with no exceptions. These rules can be checked for compliance automatically, without reference to the semantics of the text. We identified 5 rules that link to this in the annotation guide (Saurí et al., 2006).

3.1.1. Evidential Link Rule

As described in Section 2.1, EVIDENTIAL links are a type of SLINK, typically introduced by reporting and perception events such as *see*, *say*, *tell*, etc.:

(7) Katy **said** she **went** to Vegas.
(said −EVIDENTIAL→ went)

Similarly, NEG_EVIDENTIAL links are typically introduced by reporting and perception events but with negative polarity:

(8) Katy **denied** that she **went** to Vegas.
(denied −NEG_EVIDENTIAL→ went)

According to the TimeML annotation guideline (Saurí et al., 2006, p. 53), Perception events will **always** introduce SLINKs of type EVIDENTIAL or NEG_EVIDENTIAL. This rule can be checked automatically.

3.1.2. Conditional Link Rule

As discussed in Section 2.1, a CONDITIONAL link is a type of SLINK that holds between two events and is usually introduced by a signal such as *if... then*:

(9) If she **gets** the medicine, she'll **feel** better.
(gets −CONDITIONAL→ feel)

The TimeML annotation guideline (Saurí et al., 2006, p. 54) specifies that the conditional conjunctions (if-clauses) will **always** introduce CONDITIONAL SLINK. Based on the rule, we can automatically detect whether each if-clause is associated with a CONDITIONAL link.

3.1.3. Causative Event Rule

The TimeML annotation guideline (Saurí et al., 2006, p. 7), specifies that any construction of the "*subject* event + *causative* event + *object* event", **must** introduce an IDENTITY TLINK between the subject event and the causative event. For example:

(10) The **rains** **caused** the **flooding**.
(rains −IDENTITY→ caused)

For this rule, we implemented a system that identifies the "*subject* event + *causative* event + *object* event" where the causative event has a head word which is any inflected form of *cause*, then checks whether an IDENTITY TLINK is defined between the *subject* event and the *causative* event.

3.1.4. ALINK Replacement Rule

As described in Section 2.1, an ALINK occurs between an aspectual event and its argument event. The TimeML annotation guidelines (Saurí et al., 2006, p. 58) specifies that the `eventInstanceID` attribute of an ALINK must contain the ID of the aspectual event while `relatedToEventInstance` attribute indicates the ID of the argument event. For example:

(11) He **finished**$_1$ **watching**$_2$ The Office.

```
<ALINK eventInstanceID="1"
relatedToEventInstance="2"
relType="TERMINATES"/>
```

For ALINK replacement rules, we implemented a system that goes through every ALINK and checks the `eventInstanceId` to ensure compliance.

3.1.5. ALINK-SLINK Incompatibility Rule

As discussed in Section 2.1, an ALINK occurs between an aspectual event and its argument event while an SLINK occurs between a subordinated event and its related event. A TLINK can be introduced along with an ALINK. Similarly, a TLINK can also be introduced with an SLINK. However, by definition an ALINK and an SLINK cannot hold between the same pair of events. For example, in Example (7), *said* and *went* are related by an EVIDENTIAL subordination relationship. They also have an AFTER temporal relationship that indicates *said* happened after *went*. This is an instance where TLINK and SLINK can be present simultaneously. Similarly in Example (11), between *finished* and *watching* there is both a TERMINATES aspectual link and an ENDS temporal link. For this rule, we implemented a system that checks every pair of nodes in the TimeML graph for compliance.

3.2. Graph Rules

As discussed previously, a TimeML graph is a graph where the nodes are events and temporal expressions, and edges are TimeML links. TimeML graphs are useful for visualizing TimeML annotations. We evaluated the graphs in TimeBank for temporal inconsistency, graph disconnectivity, and unnecessary self-loops.

3.2.1. Temporal Inconsistency

Although developers of the TimeBank corpus claimed that the corpus is fully consistent, Derczynski and Gaizauskas (2012) found 30 inconsistent stories. However, they considered only TLINKs when computing consistency. Here we consider both TLINKs and ALINKs when computing overall consistency, because ALINKs also indicates temporal relationships.

For this evaluation method we used TLEX, the method described by Finlayson et al. (2021). TLEX is a system for producing timelines from TimeML graphs; as one of its substeps, it implements a method for checking graph consistency. TLEX proceeds as follows. First, it partitions the TimeML graph into a set of temporally connected subgraphs (subgraphs of the full graph whose nodes are connected to each other via TLINKs and ALINKs). Second, TLEX transforms temporally connected subgraphs into point algebra (PA) graphs. Then finally, it assigns real numbers for each node in the PA graph. Barták et al. (2014) showed that there is a solution (i.e., assignment of real numbers to time points) that satisfies the PA graph if and only if the graph is consistent. If the real number assignment is not possible, then the graph is inconsistent (Barták et al., 2014). Therefore, TLEX tells us whether the graph is consistent.

To illustrate, in Figure 2 we provide the PA graph of the TimeML graph shown in Figure 1, as well as the real number assignment.

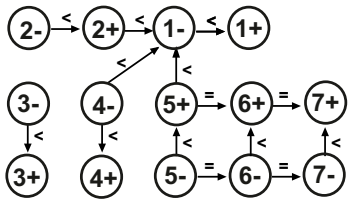

Figure 2: Visualization of the output of the transforming TimeML graph to PA graph.

3.2.2. Graph Disconnectivity

A TimeML graph extracted from a single annotated file might comprise multiple disconnected subgraphs. This is because the annotators sometimes forget to annotate TimeML links between events and temporal expressions. Therefore multiple disconnected subgraphs in a single file suggests the possibility of certain missing links, which can be manually checked. Disconnectivity can be easily detected by walking the graph. In Figure 3, we show the TimeML graph of the TimeBank file `wsj_0991.tml`. When the graphs are compared with the text, one can identify one missing link (t23−BEFORE→t19) which connects the subgraph in the upper left to the subgraph on the right. The subgraph in the lower left is correctly disconnected from the other two subgraphs.

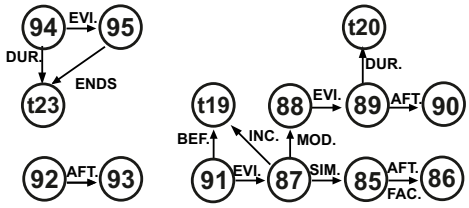

Figure 3: Visualization of a disconnected graph. The numbers indicate the event instance IDs or time IDs (starts with *t*). The file contains three disconnected graph, therefore, potentially two links are missing.

3.2.3. Redundant Self-Loops

A self-loop in a link from a node to itself, which are always incorrect in the TimeML scheme. There are two types of self-loops: inconsistent self-loops and redundant self-loops. Any ALINK or TLINK (except SIMULTANEOUS or IDENTITY) from a node to itself is an inconsistent self-loops and will be detected when evaluating the graph for inconsistency as described in Section 3.2.1. On the other hand, a redundant self-loop is a SIMULTANEOUS or IDENTITY TLINK self-loop. While they do not cause inconsistency, they also do not provide any useful information. Therefore, they are redundant.

3.3. Missing Annotations

Our final check involved comparing the gold-standard TimeBank annotations with automatically generated annotations in an attempt to find missing annotations. As described in Section 2.2, CAEVO is an automatic TimeML parser that can take raw text as an input and generate TimeML annotations. Since it has the best score on detecting events and temporal expressions, we used CAEVO to generate annotations for comparison of the TimeBank corpus.

We processed raw TimeBank texts with CAEVO, using it to detect events and temporal expressions. We then compared the output with the gold-standard annotations manually, judging whether events and times detected by CAEVO, but not in TimeBank, were correct. Because CAEVO is not perfect, not all events and times detected by CAEVO are correct; indeed, approximately half of the events detected by CAEVO but not in TimeBank were generic events, which should be excluded according to the TimeML annotation guide (Saurí et al., 2006,

p. 7). In the following example, *driving* should not be tagged as an event since it's a generic event, however, CAEVO will tag it as an event.

(12) **Driving** under influence is illegal.

For temporal expressions, CAEVO cannot correctly filter some proper names, and so labels incorrectly. In the following example, SEC represents the US Security and Exchange Commission. However, CAEVO detects SEC as a second.

(13) Through his lawyers , Mr. Antar has denied allegations in **the SEC** suit.

For the manual comparison process, two of the authors did a double annotation. First, the authors compared the events and times separately which allows us to calculate the inter-annotator agreement score as 0.67. The reason why the inter-annotator agreement score is relatively low is disagreements over whether particular events were generic, which is often a subtle judgement. Then, the annotators conferred on the annotations, examining disagreements one by one and comparing them to the text. When the annotators discovered a missing event or temporal expression to be added, they also added any missing links to connect the event or temporal expression to the rest of the graph.
We didn't compare the CAEVO-annotated links with the TimeBank links. This is because (a) CAEVO has only a 0.51 F1 score on extracting TLINKs, (b) it can only generate 5 types of TLINKs and does not generate ALINKs or SLINKs, and (c) it uses transitive closure to generate links which creates noisy graphs with a lot of extra links.

4. Results & Corrections

4.1. Results of TimeML Annotation Rules

We built a system that checks the strict TimeML annotation rules defined in Section 3.1. Table 1 lists the results for each of the five rules. As we can see in the table, there are 109 instances that violate the TimeML annotation guidelines rules.

Evidential Link Rule Our system detected 48 perception events in the corpus. Only 18 of them are involved in an EVIDENTIAL link. For the rest of the perception events, 15 of them have different types of SLINKs and the other 15 of them have no SLINKs at all. We also noted that, out of 18 perception events with EVIDENTIAL, two of them have negative polarity, meaning, they should introduce NEG_EVIDENTIAL instead of EVIDENTIAL. An example from `ea980120.1830.0071.tml` that violates the evidential link rule is shown below. The perception event is underlined and in bold.

(14) Cubans want to know what we're going to tell Americans, in many cases, what their relatives in the United States are going to **hear**. (No Evidential SLINK)

For the missing annotations of evidential link rule, we define the necessary EVIDENTIAL or NEG_EVIDENTIAL links based on polarity with the new linkID. And if an SLINK was incorrect, we corrected it with a proper SLINK. For the example, we define an EVIDENTIAL link between *hear* and *tell*.

Conditional Link Rule Our system detected 64 conditional conjunctions and 45 of them have Conditional SLINKs. The rest of the 19 conditional statements have no SLINKs at all. An example that violates this rule from `wsj_0586.tml` is as follows. We added any missing Conditional SLINKs.

(15) If you take away the outside influences, the market itself looks very cheap. (No Conditional SLINK)

Causative Event Rule Our system detected 14 "*subject* event + *causative* event + *object* event" cases and only four of them correctly introduced IDENTITY TLINK between the subject event and the causative event. One of the 10 cases that violates the rule (from `ea980120.1830.0071.tml`) is shown below. For each of the 10 cases, we added the missing Identity link between the subject event and the causative event. For the example, we define IDENTITY TLINK between *concern* and *caused*.

(16) Well, this is the eve of the Pope's **visit** to one of the last bastions of Communism anywhere in the world, and it is already **causing** enormous **expectations**. (No IDENTITY TLINK)

ALINK Replacement Rule Out of 265 ALINKs, our system took the first eventID and traced back to the annotations to see if the event is an aspectual event. The results showed that 46 of them are not aspectual events, meaning 46 ALINK violate the rule. In the following example from `WSJ900813-0157.tml`, annotators put the ALINK in the reverse order.

(17) Iraq's Saddam Hussein, his options for **ending** the Persian Gulf **crisis** growing increasingly unpleasant. (crisis −TERMINATES-> ending)

ALINK-SLINK Incompatibility Rule We checked if there are any cases that violate the rule. Our system found four violations where there were a TLINK, SLINK, and ALINK between the same two nodes. All four cases involved the verbs *launched*, *offer*, *suit*, or *bid*, in a BEGINS, FACTIVE, or INITIATES relationship. For the four examples, we removed the SLINKs to follow the rule.

(18) Acquisition has **launched** a **suit** in a Delaware court.

(19) Dow Jones **launched** the **offer** on Sept. 26.

(20) A unit of DPC Acquisition Partners **launched** a $10-a-share tender **offer** for the shares.

(21) Before the **bid** was **launched**, he sought approval to boost his Paribas stake above 10%.

Rules	Total Instances	# of Violations
EVIDENTIAL **Link**	48	30
CONDITIONAL **Link**	64	19
Causative Event	14	10
ALINK **Replacement**	265	46
ALINK-SLINK **Incompatibility**	-	4
Total		109

Table 1: Results of the TimeML Annotation Rules. The number of instances that match the rule are given in **Total Instances**, while the number of those matches that violate the rule are given in **# of Violations**.

4.2. Results of Graph Rules

We checked the graph rules that are defined in Section 3.2, and the results are summarized in Table 2.

Temporal Inconsistency We showed that 65 texts are inconsistent, roughly 1/3 of the TimeBank corpus. As we can see in the Table 2, 30 inconsistent files were caused by TLINKs, which matches the results reported by Derczynski and Gaizauskas (2012). However, a slightly greater number of inconsistencies were caused by ALINKs (35 texts), which were not investigated previously. In total, we detected 110 inconsistent subgraphs across 65 texts. Additionally, we showed that 15 of those inconsistencies were caused by inconsistent self-loops which can be easily fixed by simple removal of the self-loops. Other inconsistencies required a more careful comparison of the annotations with the text. An example of an inconsistency from `wsj_1011.tml` is as follows.

(22) [**DCT:10/26/1989**$_{t57}$] The latest **results**$_{ei2048}$ include a $2.6 million one-time payment from a "foreign entity."
t57−BEFORE−>ei2048
ei2048−BEFORE−>t57

Graph Disconnectivity Our system showed that only 35 texts have a single fully connected graph. The remainder of the files contained anywhere between 2 and 34 disconnected subgraphs, suggesting the same number of potentially missing links, for an overall total of up to 739 missing links. Upon manual inspection, we found only 625 missing links, with the remaining 108 subgraphs being correctly disconnected from the other graphs in the file because of the lack of temporal information in the text. Note that Derczynski and Gaizauskas (2012) reported all files were disconnected, however, they did not consider ALINKs and SLINKs in their system. We also found 65 singleton nodes (all temporal expressions) that had no incoming or outgoing links, all of which should have been connected to some other node in the file.

To resolve some disconnections, we implemented a system that automatically suggests links between temporal expressions in disconnected subgraphs. To do that, our system selects a temporal expression from the main graph (timex-1) and from the subgraph (timex-2), then, it normalizes their value. Finally, it creates a TLINK based on their value such as timex-1 $-$BEFORE$\rightarrow$ timex-2. This allows us in many cases to achieve connectivity automatically and correctly.

Redundant Self-loops Our system detected 10 redundant self-loops. Redundant self-loops can be removed automatically.

4.3. Results of Missing Annotations

We processed the raw TimeBank texts with CAEVO, which detected 947 events not present in the TimeBank annotations. Then, we performed the double annotation process (described in Section 3.3) and we identified 317 missing events in the TimeBank corpus. The reason for the significant difference between the number of detected events and those determined to be correct is that (a) CAEVO is not perfect (0.81 F_1 score in event detection) and (b) CAEVO is especially weak in detection of generic events, which are explicitly excluded by the TimeML scheme.

For temporal expressions, CAEVO detected 93 different temporal expressions not present in TimeBank. After the manual annotation process, we determined 52 of these were correct. The main reason for 41 false positive temporal expressions is that CAEVO detected some proper names as temporal expression such as "the **SEC**", "USA **Today**", etc.

Each missing event or temporal expression also potentially implied missing links to connect the node to the main graph. We added the missing links based on our understanding of text. For all corrections, we strictly follow the TimeML annotation guidelines rules. The total number of each correction is shown in Table 3.

Because TimeBank 1.2 is distributed under license by the LDC, we do not directly provide the corrected corpus. Instead, we provide patch files that can be applied to the original TimeBank 1.2 using the standard Linux or Unix `diff` command.

5. Discussion

In this paper, we presented a semi-automatic evaluation of the TimeBank 1.2 corpus. We showed that TimeBank has 1,630 incorrect or missing annotations, for which we provide corrections. These misannotations are mainly caused by manual annotation mistakes. And there are many other gold-standard TimeML corpora that might suffer from the manual annotation process. The complexity of the schema suggests that any gold-standard TimeML annotations should, as a best practice, use such an evaluation approach to ensure the highest quality annotations.

Rules	Events	Times	TLINK	ALINK	SLINK	Total
TimeML Rules	-	-	10	46	53	109
Graph Rules	-	-	722	61	-	783
Missed Annotations	317	52	369	-	-	738
Total Corrections	317	52	1,101	107	53	1,630
Total # Objects in TimeBank 1.2	7,935	1,414	6,418	265	2,932	18,964
% Total Corrected	3.9%	3.7%	17.2%	40.8%	1.9%	8.6%
Total # Objects in Corrected TimeBank	8,252	1,466	7,351	271	2,982	20,322

Table 3: Summary of corrections to the TimeBank 1.2 corpus, split into different categories.

Rules	# of Files	# of Violations
Inconsistency (TLINKs)	30	38
Inconsistency (all)	65	110
Disconnectivity	148	625
Redundant Self-loops	9	10
Total		783

Table 2: Results of checking the Graph Rules. The number of files that contained a violate of the rule is given in **# of Files**, while the number individual violations across all files is given in **# of Violations**.

6. Contributions

Our contributions in this paper are three-fold. First, we presented a comprehensive evaluation of the TimeBank corpus comprising 10 different automatic or semi-automatic checks. Second, we showed that TimeBank corpus has 1,630 incorrect or missing annotations. Finally, we provided corrections for the all TimeBank corpus incorrections and we released patch files as well as our code for use by other researchers in the field[2].

7. Acknowledgements

We gratefully acknowledge valuable discussions with Akul Singh and Emmanuel Garcia. This work was partially funded by research grants TO-134841, TO-135998, and TO-139837 to FIU from MITRE Corporation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of MITRE Corporation. This work was also partially supported by DARPA Award HR001121C0186 under the INCAS Program.

8. Bibliographical References

Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, November.

Bansal, R., Rani, M., Kumar, H., and Kaushal, S. (2018). Temporal event detection using supervised machine learning based algorithm. In *International Conference on Innovations in Bio-Inspired Computing and Applications*, pages 257–268. Springer.

Barták, R., Morris, R., and Venable, K. (2014). *An Introduction to Constraint-Based Temporal Reasoning*. Morgan & Claypool Publishers.

Bethard, S. (2013). Cleartk-timeml: A minimalist approach to tempeval 2013. In *Second joint conference on lexical and computational semantics (* SEM), volume 2: proceedings of the seventh international workshop on semantic evaluation (SemEval 2013)*, pages 10–14.

Boguraev, B. and Ando, R. K. (2006). Analysis of TimeBank as a resource for TimeML parsing. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May. European Language Resources Association (ELRA).

Boguraev, B., Pustejovsky, J., Ando, R., and Verhagen, M. (2007). Timebank evolution as a community resource for timeml parsing. *Language Resources and Evaluation*, 41(1):91–115.

Caselli, T. and Morante, R. (2018). Systems' agreements and disagreements in temporal processing: An extensive error analysis of the tempeval-3 task. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Chambers, N., Cassidy, T., McDowell, B., and Bethard, S. (2014). Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Chambers, N. (2013). Navytime: Event and time ordering from raw text. Technical report, NAVAL ACADEMY ANNAPOLIS MD.

Chang, A. and Manning, C. D. (2013). Sutime: Evaluation in tempeval-3. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 78–82.

Chen, S., Wang, G., and Karlsson, B. (2019). Exploring word representations on time expression recognition. Technical report, Technical report, Microsoft Research Asia.

Derczynski, L. and Gaizauskas, R. (2012). Analysing

[2]Code and data may be downloaded from `https://doi.org/10.34703/gzx1-9v95/EFNL6H`.

temporally annotated corpora with cavat. *arXiv preprint arXiv:1203.5051*.

Färber, M. and Rettinger, A. (2015). Toward real event detection. In *derive@ ESWC*, pages 24–34.

Finlayson, M. A., Cremisini, A., and Ocal, M. (2021). Extracting and aligning timelines. *Computational Analysis of Storylines: Making Sense of Events*, page 87.

Inel, O. and Aroyo, L. (2019). Validation methodology for expert-annotated datasets: Event annotation case study. In *2nd Conference on Language, Data and Knowledge (LDK 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Kadir, M., Sobhan, S., and Islam, M. Z. (2016). Temporal relation extraction using apriori algorithm. In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, pages 915–920. IEEE.

Kolomiyets, O. and Moens, M.-F. (2009). Comparing two approaches for the recognition of temporal expressions. In *Annual Conference on Artificial Intelligence*, pages 225–232. Springer.

Mani, I., Verhagen, M., Wellner, B., Lee, C., and Pustejovsky, J. (2006). Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 753–760.

Mirroshandel, S. A. and Ghassem-Sani, G. (2011). Temporal relation extraction using expectation maximization. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 218–225.

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

Ning, Q., Yu, Z., Fan, C., and Roth, D. (2018). Exploiting partially annotated data for temporal relation extraction. *arXiv preprint arXiv:1804.08420*.

Ocal, M. and Finlayson, M. (2020). Evaluating information loss in temporal dependency trees. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2148–2156, Marseille, France, May. European Language Resources Association.

Ocal, M., Perez, A., Radas, A., and Finlayson, M. (2022). Holistic evaluation of automatic timeml annotators. In *Proceedings of The 13th Language Resources and Evaluation Conference*, Marseille, France, June. European Language Resources Association.

Pustejovsky, J., Castano, J. M., Ingria, R., Sauri, R., Gaizauskas, R. J., Setzer, A., Katz, G., and Radev, D. R. (2003a). Timeml: Robust specification of event and temporal expressions in text. *New directions in question answering*, 3:28–34.

Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., and Lazo, M. (2003b). The timebank corpus. *Proceedings of Corpus Linguistics*, 01.

Saurí, R., Littman, J., Knippen, B., Gaizauskas, R., Setzer, A., and Pustejovsky, J. (2006). Timeml annotation guidelines. *Version*, 1(1):31.

UzZaman, N. and Allen, J. F. (2010). Extracting events and temporal expressions from text. In *2010 IEEE Fourth International Conference on Semantic Computing*, pages 1–8. IEEE.

UzZaman, N., Llorens, H., and Allen, J. (2012a). Evaluating temporal information understanding with temporal question answering. In *2012 IEEE Sixth International Conference on Semantic Computing*, pages 79–82. IEEE.

UzZaman, N., Llorens, H., Allen, J., Derczynski, L., Verhagen, M., and Pustejovsky, J. (2012b). Tempeval-3: Evaluating events, time expressions, and temporal relations. *arXiv preprint arXiv:1206.5333*.

Verhagen, M., Mani, I., Saurí, R., Littman, J., Knippen, R., Jang, S. B., Rumshisky, A., Phillips, J., and Pustejovsky, J. (2005). Automating temporal annotation with tarsqi. In *ACL*, pages 81–84.

Veyseh, A. P. B., Nguyen, M. V., Min, B., and Nguyen, T. H. (2021). Augmenting open-domain event detection with synthetic data from gpt-2. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 644–660. Springer.

Zhong, X., Sun, A., and Cambria, E. (2017). Time expression analysis and recognition using syntactic token types and general heuristic rules. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 420–429.