When it's all piling up: investigating error propagation in an NLP pipeline

Tommaso Caselli, Piek Vossen, Marieke van Erp, Antske Fokkens, Filip Ilievski, Ruben Izquierdo Bevia, Minh Le, Roser Morante, and Marten Postma

Computational Lexicology and Terminology Lab
The Network Institute
VU University Amsterdam
cltl-vu@googlegroups.com

Abstract. We present an analysis of a high-level semantic task, the construction of cross-document event timelines from SemEval 2015 Task 4: TimeLine, to trace down errors to the components of our pipeline system. Event timeline extraction requires many different Natural Language Processing tasks among which entity and event detection, coreference resolution and semantic-role-labeling are pivotal. These tasks yet depend on other low-level analysis. This paper shows where errors come from and whether they are propagated through the different layers. We also show that performance of each of the subtasks is still insufficient for the complex task considered. Finally, we observe that there is not enough semantics and inferencing within the standard NLP techniques to perform well.

Keywords: NLP, error analysis, temporal event ordering

1 Introduction

Textual interpretation requires many analyses ranging from tokenization, PoS-tagging to high-level semantic tasks such as entity-detection or semantic-role-labeling (SRL). In Natural Language Processing (NLP), such tasks are usually considered in isolation. However, the output of low-level analyses is used to achieve higher levels of interpretations and it is well-known that their errors propagate to higher levels. Furthermore, different modules may make incoherent claims on the same text elements. A Named-Entity Recognition and Classification (NERC) system may say that *Ford* is an entity of the type `person`, whereas the entity linker finds the URI to the `company` and the SRL module assigns the role of `vehicle`. Typical NLP architectures allow dependencies in one direction only and do not have any mechanism to reconcile inconsistency. Therefore most high-level modules are not designed to detect inconsistency and simply ignore competing results from other modules. Especially when high levels of semantic processing and interpretation are the goal, it becomes very difficult to relate the performance of the system to the performance of each of the sub-modules and to improve it.

In this paper, we present an error analysis of a complex semantic task: the SemEval 2015 Task 4: TimeLine: Cross-Document Event Ordering[1] to learn more about the dependencies between modules and missed opportunities. The most relevant NLP sub-tasks for timeline extraction are entity and event detection, coreference resolution, semantic-role-labeling and time expression recognition and normalization. Timelines are created by combining the results of 10 modules carrying out subtasks. This pipeline is an excellent case for demonstrating the complexity of the process and the (non-)dependencies between the modules. Our analysis starts from the errors creating the timelines and drills down to lower levels that provide the necessary elements. We show that error propagation from lower levels occurs, but its impact remains limited. Errors from high level tasks piling up form the main cause of overall low performance. Our analyses reveal that several errors can be avoided if information from various modules is integrated better. However, only full comprehension of the context could yield high results on this task.

Our study differs from previous work in that we quantify error dependencies and shortcomings of a pipeline system rather than providing users with tools for error tracking or component integration. To the best of our knowledge, the most similar work we have identified in literature is Clark et al. (2007) [5], where the authors aimed at identifying what type of lexical knowledge is required to tackle the Textual Entailment Task.

The paper is further structured as follows. In Section 2 we describe the task and the NLP pipeline that we used to participate. We give a detailed trace-back analysis for the errors of our system in Section 3. We discuss these errors further in Section 4 after which we conclude in Section 5.

2 Timeline Extraction

The SemEval 2015 Task 4: TimeLine: Cross-Document Event Ordering aims at identifying all events in a set of documents where a given entity plays the PropBank [14] role Arg0 or Arg1. These events are then anchored to dates, where possible, and ordered chronologically so as to create an *entity-centric timeline*. A full description of the task can be found in Minard et al. 2015 [13].

Figure 1 illustrates our pipeline indicating the dependencies among the different modules.[2] The pipeline modules mostly represent state-of-the-art systems, some developed by third parties, and integrated in the NewsReader project.[3] The rest of this section describes the main NLP modules of our system.

2.1 Event Extraction and Entity-linking

Event extraction and entity-linking is based on the output of the SRL module of our pipeline. The predicates identified by the SRL module form candidate events.

[1] `http://alt.qcri.org/semeval2015/task4/`

[2] A detailed description can be found in Agerri et al. (2014) [1].

[3] `http://www.newsreader-project.eu`

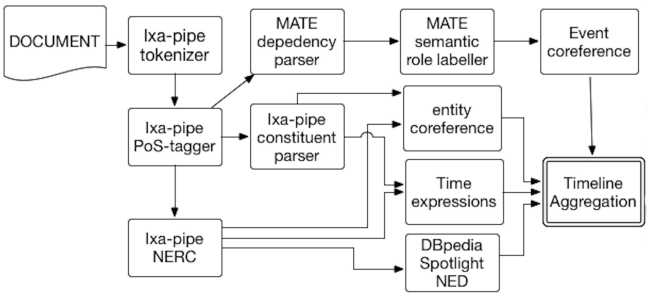

Fig. 1. Overview of the NewsReader pipeline

The SRL module outputs PropBank roles, so we can directly identify relevant event-entity links by selecting the Arg0 and Arg1 roles of the predicates.

The SRL module we use is based on the MATE-tools [3]. It takes tokens and PoS-tags (which are used as features) as input and performs dependency parsing and SRL in one joint step.

2.2 Entity Identification and Coreference

The next step is to identify which events have an Arg0 or Arg1 that corresponds to the target entity of a timeline. Three modules are involved in this sub-task: a) NERC to identify and classify names, b) a NED to disambiguate named entities by means of DBpedia URIs and c) coreference resolution to identify which other expressions in the text have the same referent.

The NERC module is a supervised machine learning system trained on the CoNLL 2002 and 2003 shared tasks [18, 19]. It takes tokens, lemmas and PoS tags as input and uses local features only. The outcome of the classifier is taken as input for the NED module which is based on DBpedia spotlight.

The coreference resolution module is a reimplementation of Stanford's Multi Sieve Pass system [11]. It is a rule-based system that uses lemmas, PoS-tags, the identified entities and constituents.

2.3 Event Coreference

The event coreference module uses the predicates from the SRL layer. Predicates with the same lemma in the same document are considered coreferential as well as predicates with high similarity in WordNet [7].

2.4 Time Expressions

We use FBK TimePro to identify time expressions in text. This machine learning system is trained on TempEval3 data [20]. Time expressions are consequently normalized using the timenorm library [2], which is integrated in the system. FBK TimePro uses tokens, lemmas, PoS-tags, the entities identified by the NERC module and constituents.

2.5 Aggregating Timelines

Each of the above modules generates an independent stand-off layer of annotation in the NLP Annotation Format (NAF, [8]), which needs to be combined to further generate timelines. For this purpose, the mentions of identical entities and identical events are combined in a single representation, where the events are anchored to time expressions or to the document publication date. Event anchoring is based on a baseline system that links events to temporal expressions mentioned in the same sentence or in one of the two preceding or one following sentence. For all matched mentions of an entity (including coreferential phrases), we consider the roles of all mentions of events in which they are involved and output the event with the time-anchor or no time-anchor in the assumed proper time-order. Likewise, our system potentially detects events for entities such as *Airbus* in cases where the role is described through coreferential expressions, such as *they* or *the plane*. In Table 1 we report the performance of our system in the SemEval task and for four target entities[4].

Table 1. System Results - overall and for target entities.

Entity	Micro-P	Micro-R	Micro-F
All	4.806	13.732	7.120
Airbus	8.547	16.363	11.228
Airbus A380	4.31	7.5	5.479
Boeing	11.718	18.279	14.281
Boeing 777	0.0	0.0	0.0

Overall, performance is very low and varies a lot across the 4 entities. Although it is a very difficult task, a deep error analysis is required to find the main causes for this performance, which will be presented in the next section.

3 Error Analysis

To get more insight in the problems for this task, we want to answer the following questions:

- *Which modules are responsible for most errors?*
- *How are errors propagated to other modules in the pipeline?*
- *To what extent do different modules make different statements on the same interpretation?*

All of our modules have been benchmarked against standard datasets obtaining state-of-the-art performances.[5] However, what these figures do not tell us is how errors propagate and whether a certain number of errors in different modules make it impossible to derive the high-level structure of a timeline.

We perform an in-depth error analysis of four target entities (Airbus, Airbus380, Boeing and Boeing 777) by reversing the steps in the pipeline to find the module that breaks the result. We used false negatives (FN) and false positives (FP) errors as starting point.

[4] More details on the system can be found in Caselli et al. (2015) [4]
[5] For more details on the benchmarking see Erp et al. (2015) [6].

3.1 Events

The analysis of events involved a total of 178 events for the FNs of all four target entities and 209 events for the FPs of Airbus and Airbus A380. These two entities are paradigmatic cases, i.e., the set of detected errors will apply to the other entities as well.

We identified 21 error types: 9 error types are due to specific annotation layers (e.g. SRL or Event_detection), 9 are due to a combination of annotation layers (e.g. SRL + Entity_COREFERENCE), and 3 are not related to the pipeline (e.g. Gold data or spelling errors). Table 2 provides an overview of the errors.

Table 2. False Negatives and False Positives for events in the timelines.

Error Type	False Negatives				False Positives	
	Airbus	Airbus A380	Boeing	Boeing 777	Airbus	Airbus A380
Entity_COREFERENCE	7	22	5	2	2	0
Event_COREFERENCE	1	0	1	0	35[a]	4
Event_COREFERENCE + Event_filter	0	0	0	0	1	1
Event_COREFERENCE + TML_match	0	0	0	0	6	2
Event_filter	0	0	0	0	1	0
TML_match	0	0	2	0	3	0
Final_Entity_Extraction	0	0	0	0	0	77
Event_detection	6[b]	12[c]	20[d]	1	13	2
Event_detection + Final_Entity_Extraction	0	0	0	0	0	23
Event_detection + Event_COREFERENCE	0	0	0	0	18	2
Event_detection + Entity_COREFERENCE	0	3	0	0	0	0
SRL	10	3	10	0	1	5[e]
SRL_implicit_arg	5	10	5	0	0	0
SRL + Entity_COREFERENCE	1	6	1	2	0	0
SRL + Event_COREFERENCE	0	0	0	0	1	0
SRL + NE	0	3	0	2	0	0
NE	2	1	10	2	0	0
NE + Final_Entity_Extraction	2	10	1	0	0	0
Gold_Error	1	1	1	0	2	2
Document_Id	0	0	1	0	0	0
Spelling	0	0	1	0	0	0

False Negatives: The analysis has highlighted three major sources of errors all related to high-level semantic tasks, namely Event_detection (39 events); Entity_COREFERENCE (36 events) and SRL (23 events).

Error propagation from low-level annotation layer affects event detection for only 8 cases (i.e. wrong PoS tagging). The majority of cases is directly affected by the SRL layer with 11 instances of partial detection (i.e. multi-tokens events) and 24 cases of events realized by nouns.

Entity coreference affects event extraction indirectly as the target events are detected but they cannot be connected to the target entities.

Finally, the SRL module mainly introduces two types of errors: wrong role assignment or implicit arguments in the predicate structure.

[a] It contains 1 case due to wrong PoS tagging.
[b] It contains 2 cases of partial detection and 1 case due to wrong PoS tagging.
[c] It contains 6 cases of partial detection and 3 cases due to wrong PoS tagging.
[d] It contains 3 cases of partial detection and 3 cases due to wrong PoS tagging.
[e] All cases include also errors with Final_Entity_Extraction.

False Positives: These errors point to issues related to over-generation of events and of event entity linking. Most errors concern Event_COREFERENCE (39 cases, plus 10 cases in combination with other modules) and Event_detection (15 cases, plus 43 in combination with other modules).

The event coreference errors point to both a limit of the specific annotation layer and to a propagation of errors from the SRL module.

The SRL module responsible for event detection does not only miss most nominal events, it also overgenerates them. This is due to the implementation of the SRL module which labels all entries in NomBank [9] as events. This assumption is valid in a broad perspective on argument structures, but not all nouns with argument structures have an event reading.

The Final Entity Extraction refers to errors in entity matching for the final timeline. The high number of errors for Airbus A380 (77) is caused by the substring method used as a back-off strategy.

3.2 Semantic Role Labeling

We analyzed the SRL for 60 events in the gold timeline for the entity *Airbus*. Table 3 provides an overview of our findings.

Table 3. SRL statistics on the Airbus timeline.

Events in gold timeline (gold events)	60
Gold events in the SRL layer	54
Gold events in out-of-competiton timeline	25
Gold events not in out-of-competiton timeline	29
Correct SRL assignment in SRL layer	41
Correct SRL assignment not in out-of-competiton timeline	19
Correct SRL assignment in out-of-competiton timeline	22
Incorrect SRL assignment in SRL layer	15
Incorrect SRL assignment not in out-of-competiton timeline	10
Incorrect SRL assignment in out-of-competiton timeline	5
Events in out-of-competiton timeline	103

The SRL layer contained 54 out of 60 gold events. From these 54 events, 41 had a correct SRL assignment. However, only 22 out of these 41 events ended up in the system's timeline. We distinguish two groups among the 19 cases of correct SRL that did not appear in the timeline:

Entity in explicit semantic role (11 cases) We found 4 cases where the NED system failed to link the named entity *Airbus* to the correct DBpedia URI. In 2 cases, *Airbus* was not recognized as the head of the NP. In the other 5 cases, the error is caused by the coreference module.

Entity in implicit semantic role (8 cases) Implicit roles [10, 17] are roles that are not instantiated in the clause, but can be recovered from the context. In 4 cases, the role could be inferred from information in the same sentence, and 4 cases required previous context. These 8 errors clearly originate in the SRL layer but are not errors of the SRL module, since the SRL assignment is correct according to the state-of-the-art tools.

3.3 Entities

Named entities can be mentioned explicitly in text or implicitly through coreferential chains. We analyzed both cases by inspecting the entities which appear in the roles Arg0 and Arg1 of the event predicates. Explicitly mentioned named entities are analyzed using the following steps:

1. Did disambiguation (NED) fail or recognition (NERC)?
2. If disambiguation failed, is the entity recognition incorrect? This includes both missed entities and incorrect spans (e.g. *Airbus to* instead of *Airbus*)
3. If recognition failed, are there problems with the PoS, lemmas or tokens?

For each coreferencial mention, we traced back the correct coreference chain(s). We identified two types of errors: missing coreference link and wrong coreference link. For each case, we also inspected whether the coreferential problem is caused by a wrong PoS tag. Overall, 16 cases were identified where the anaphoric mention was realized by a pronoun not linked to the correct antecedent. In 20 cases, the anaphoric element was realized by a Full Definite Noun Phrase (FDNP) of the form "the + (modifier) + noun". An in-depth analysis of these cases, in the line of Poesio and Vieira (1998) [15], showed that to solve such coreferential chains world knowledge related to the target entities is required. For instance, to identify that *the aircraft* corefers with its antecedent *Airbus A380*, knowledge about the fact that *Airbus A380* is an (instance of) of an aircraft is required.

Results are reported in Table 4. The numbers for coreference relations only concern direct links between a Named Entity and an anaphoric mention. Cases of implicit arguments, or zero anaphora, are not reported in the table.

Table 4. Analysis of the entity errors.

Error Type	False Negatives				False Positives	
	Airbus	Airbus A380	Boeing	Boeing 777	Airbus	Airbus A380
NE disambiguation error (overall)	4	14	11	4	0	0
NE recognition error	2	14	0	4	0	0
Wrong part of speech	0	3	0	4	0	0
Wrong lemma	0	1	0	0	0	0
Unresolved coreference (overall)	8	31	6	4	2	0
Wrong coreference link	4	3	2	1	2	0
Wrong part of speech	0	0	1	1	0	0

3.4 Time expressions

Normalized time expressions are needed to anchor and order events on the timeline. We analyzed the system's performance on the events in the gold timelines as well as the events identified by our pipeline for Airbus and Airbus 380.

The gold standard of Airbus and Airbus A380 contains 103 events that need to be linked to a specific date. The system has no mechanism for disambiguation and extracted 208 links. There were 21 extracted events linked to a specific date and 91 links found. Unsurprisingly, time anchoring has decent recall and low

Table 5. Performance and error analysis of time anchoring

Entity	correct	wrong span	wrong year	other event	missing link	other interpretation	coreference
Airbus (gold)	25	19	11	27	13	7	1
A380 (gold)	36	13	26	38	13	8	5
Airbus (extracted)	10	5	6	12	0	3	0
A380 (extracted)	5	0	5	11	2	2	0

precision: Recall was 59.2% on gold events and 71.4% on the extracted events. Precision was 30.5% on gold and only 16.4% on the extracted set.

Table 5 provides an overview of the outcome of our analysis. The first column provides the number of correctly identified dates. We found five different errors and a handful of cases where the date should be identified by event coreference (i.e. not a time anchoring error). The first two errors are related to identifying the correct date. FBK TimePro identified the wrong span leading to 32 errors. These errors occur despite the constituents being correct. The error propagates up to timenorm which provides the wrong interpretation of the date. Timenorm furthermore systematically places underspecified dates in the past, so that all future events are placed in the wrong year (37 errors).

The next two errors concern the correct link between a date and an event. Finally, there are interpretation errors, where event and date are indeed connected, but the relation is not an anchoring one (but, e.g., a preceding relation). Among them, 12 cases can only be solved by world knowledge or interpretation of context. Multiple errors can occur around the same date or event, e.g., a date may be placed in the wrong year and be wrongly linked to the event.

Overall, the system is quite good at identifying time expressions and interpreting them. The main challenge lies in linking events with the correct date. This is also reflected by 13 cases where the Gold Standard seemed incorrect. With everything piling up, our system performs much worse with only 15 out of 108 events identified and anchored correctly.

4 Discussion

The analyses show that error propagation is an issue which mainly starts from high-level tasks such as entity detection and linking, event detection and SRL. The impact of error propagation from low-level layers appears to be minimal in our data. For example, we noted that the PoS-tagger correctly detects proper nouns, but the NERC module still fails.

Another conclusion is that modules require more semantic analysis and inferencing. This specifically applies to coreference and implicit arguments of events. Possibly, this requires richer semantic resources and broadening the context of analysis rather than limiting it to local and structural properties. In many cases, participants of events are mentioned in other sentences than the one where the event occurs, which is missed by sentence-based SRL. Semantics can also exploit word-sense-disambiguation (WSD) for coreference and event detection. For event coreference, the highest similarity is currently taken across all meanings of predicates, connecting the same predicates in different meanings across the

text. Restricting the similarity to dominant meanings appears to eleminate false positives of event coreference links with higher f-measures. The opposite holds for nominal coreference that fails to find bridging relations across NPs and entities (false negatives) due to lack of semantic connectivity. Widening reference to capture metonymy, e.g. *Airbus A380* implies *Airbus*, requires more elaborate semantic modeling. With respect to event detection, WSD can eliminate false positive nominals that are not events in the dominant meaning.

Finally, temporal processing must be improved, with a particular focus on temporal anchoring. Time expressions are sparse, but they are the most reliable source of information for anchoring events and provide partial ordering. Additional research based on the notion of temporal containers [16] is needed as temporal anchoring remains challenging even with perfect detection. Richer temporal resources are needed to improve the detection of explicit and implicit temporal relations between events. The order in which events are mentioned can provide a baseline, but its performance depends on the text type.

5 Conclusion and Future Work

We provided a detailed error analysis of a NLP pipeline system that took part in the SemEval 2015 Task 4: TimeLine. We have shown how the system's poor performance is in part due to errors in different modules of the pipeline (e.g. SRL, NERC and NED) and partially to lack of rich semantic models in current state-of-the-art systems.

An interesting aspect is the relatively low error propagation from low-levels to high-levels. This shows that most basic tasks such as tokenization, PoS-tagging and lemmatization achieve reliable performance and could almost be considered as solved. We say *almost*, because errors in these layers do propagate to high-level layers (also see Manning (2011) [12]).

Semantic tasks, such as NERC, NED, event detection, SRL and temporal processing, are far from being solved. The mismatch in performance between benchmark datasets and the task points out that more research and new solutions are required. We found evidence that some sub-tasks are deeply interrelated (e.g. event detection and SRL; NERC, NED and entity coreference) and require better integration to boost performance. Furthermore, some errors can only be avoided by rich semantic models interpreting context.

The complex task of timeline construction will first of all greatly benefit from dedicated modules for event detection: the presence of an argument structure does not necessarily correlate with event readings. Secondly, coreference resolution needs to be improved through more elaborate semantic modeling. Finally, no module uses the output of the WSD module despite evidence that a more optimal selection of senses will help their task. Future work will concentrate on these pending issues.

References

1. Agerri, R., Aldabe, I., Beloki, Z., Laparra, E., de Lacalle, M.L., Rigau, G., Soroa, A., Fokkens, A., Izquierdo, R., van Erp, M., Vossen, P., Girardi, C., Minard, A.L.: Event detection, version 2. NewsReader Deliverable 4.2.2 (2014)
2. Bethard, S.: A Synchronous Context Free Grammar for Time Normalization. In: Proceedings of EMNLP 2013. pp. 821–826 (2013)
3. Björkelund, A., Bohnet, B., Hafdell, L., Nugues, P.: A high-performance syntactic and semantic dependency parser. In: Proceedings of the 23rd COLING: Demonstrations. pp. 33–36 (2010)
4. Caselli, T., Fokkens, A., Morante, R., Vossen, P.: SPINOZA_VU: An NLP Pipeline for Cross Document Timelines. In: Proceedings of SemEval-2015. pp. 786–790 (2015)
5. Clark, P., Harrison, P., Thompson, J., Murray, W., Hobbs, J., Fellbaum, C.: On the role of lexical and world knowledge in rte3. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. Prague (June 2007)
6. Erp, M., Vossen, P., Agerri, R., Minard, A.L., Speranza, M., Urizar, R., Laparra, E.: Annotated data, version 2. NewsReader Deliverable 3.3.2 (2015)
7. Fellbaum, C.: WordNet. Wiley Online Library (1998)
8. Fokkens, A., Soroa, A., Beloki, Z., Ockeloen, N., Rigau, G., van Hage, W.R., Vossen, P.: NAF and GAF: Linking linguistic annotations. In: Proceedings 10th Joint ISO-ACL SIGSEM Workshop. p. 9 (2014)
9. Gerber, M., Chai, J., Meyers, A.: The Role of Implicit Argumentation in Nominal SRL. In: Proceedings of the NAACL HLT 2009 (2009)
10. Gerber, M., Chai, J.: Semantic Role Labeling of Implicit Arguments for Nominal Predicates. Computational Linguistics 38(4), 755–798 (2012)
11. Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., Jurafsky, D.: Stanford's multi-pass sieve coreference resolution system at the CoNLL-2011 shared task. In: Proceedings of CoNLL 2011: Shared Task. pp. 28–34
12. Manning, C.D.: Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In: Proceedings CICLing'11 - Volume Part I. pp. 171–189. Springer-Verlag, Berlin, Heidelberg (2011)
13. Minard, A.L., Speranza, M., Agirre, E., Aldabe, I., van Erp, M., Magnini, B., Rigau, G., Urizar, R.: SemEval-2015 Task 4: TimeLine: Cross-Document Event Ordering. In: Proceedings of SemEval-2015. pp. 777–785 (2015)
14. Palmer, M., Gildea, D., Kingsbury, P.: The Proposition Bank: A Corpus Annotated with Semantic Roles. Computational Linguistics Journal 31(1) (2005)
15. Poesio, M., Vieira, R.: A corpus-based investigation of definite description use. Computational linguistics 24(2), 183–216 (1998)
16. Pustejovsky, J., Stubbs, A.: Increasing informativeness in temporal annotation. In: Proceedings of the 5th Linguistic Annotation Workshop. pp. 152–160 (2011)
17. Ruppenhofer, J., Lee-Goldman, R., Sporleder, C., Morante, R.: Beyond sentence-level semantic role labeling: linking argument structures in discourse. Language Resources and Evaluation 47(3), 695–721 (2013)
18. Tjong Kim Sang, E.F.: Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In: Proceedings of CoNLL 2002. pp. 142–147 (2002)
19. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In: Proceedings of CoNLL 2003. pp. 142–147 (2003)

20. UzZaman, N., Llorens, H., Allen, J., Derczynski, L., Verhagen, M., Pustejovsky, J.: SemEval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In: Proceedings of SemEval-2013. pp. 1–9 (2013)