Conceptual and Practical Steps in Event Coreference Analysis of Large-scale Data

Fatemeh Torabi Asr[1], **Jonathan Sonntag**[2], **Yulia Grishina**[2] **and Manfred Stede**[2]
[1]MMCI Cluster of Excellence, Saarland University, Germany
fatemeh@coli.uni-saarland.de
[2]Applied Computational Linguistics, University of Potsdam, Germany
sonntag|grishina|stede@uni-potsdam.de

Abstract

A simple conceptual model is employed to investigate events, and break the task of coreference resolution into two steps: semantic class detection and similarity-based matching. With this perspective an algorithm is implemented to cluster event mentions in a large-scale corpus. Results on test data from AQUAINT TimeML, which we annotated manually with coreference links, reveal how semantic conventions vs. information available in the context of event mentions affect decisions in coreference analysis.

1 Introduction

In a joint project with political scientists, we are concerned with various tasks of indexing the content of a large corpus of newspaper articles. To supplement other NLP tools and as an interesting information for the political scientists by itself, we are interested in keeping track of discussions around headline events such as attacks and crises. The main challenges in the project include:

1. proposing a definition of event identity, and
2. finding the actual mentions in natural text,

to construct clusters of, so-called, coreferential events. We refer to the former task as a *formal convention*, a vital step in order for useful results to be delivered to the human text analysts. The latter is basically an information extraction task once a clear problem specification is obtained.

The main objective of the paper is to shed light on each of the above tasks by applying a three-layer event ontology[1]. Terminologies from earlier theories (Davidson, 1969) up until recent work (Hovy et al., 2013a) are combined to draw an integrated picture of the event coreference problem. The semantic layer is established with the help of WordNet synsets. Related entities and timestamps are considered as fundamental event attributes that in practice can be resolved from the context of a mention. We implement an incremental event clustering algorithm with respect to the adapted ontology of events and use a minimal linguistic procedure to extract values from text for every event attribute. This system is being developed to work within a pipeline annotation project where incremental clustering performs efficiently on large-scale data.

In order to evaluate our proposed method, we have manually annotated a random selection of event mentions in the AQUAINT TimeML corpus (UzZaman et al., 2013). Performance of the automatic system in pair-wise coreference resolution is comparable to that of more sophisticated clustering methods, which at the same time consider a variety of linguistic features (Bejan and Harabagiu, 2010). The differences between the human annotator pair-wise decisions and the output of our clustering algorithm reveal interesting cases where coreference labeling is performed based upon the adapted semantic convention rather than information available in the text about time, location and participants of an event instance. In the following, we provide an overview of the adapted ontology, background on event coreference, and finally our implementation and experiments within the proposed framework on real data as well as the annotated corpus. We point to related work at the various appropriate places in the paper.

2 An Object Oriented Ontology

The general impression one gets by a review of the coreference literature, is that at the semantic

[1]The term ontology is used to refer to a conceptual model of events and connections between them rather than a particular knowledge base implementation.

Proceedings of the 2nd Workshop on EVENTS: Definition, Detection, Coreference, and Representation, pages 35–44,
Baltimore, Maryland, USA, June 22-27, 2014.

formalism level, events are engaged with a higher degree of complexity and more variety than entities. That is probably because of the concrete nature of entities: intuitively, an event *happens*, whereas, an entity *exists*. As a subject matter, the latter is more straightforward to get decomposed into smaller components and be identified by certain feature attributes. The ontology explained in this chapter is general in the sense that one could (perhaps should) start understanding it by examples about entities.

A realized entity belongs to a class of entities sharing the same set of attributes. For example, president Obama, as long as being talked in a political context is considered as an instance of the class `PRESIDENT`, comprising attributes such as `Country`, `Party` and `Duration` of presidency. Any other president can be compared against Obama, with respect to the attribute values associated with them. Therefore, Bush is a different instance of the class `PRESIDENT` regarding the fact that a different political `Party` as well as a different presidential `Duration` are assigned to him. Detecting mentions of these `PRESIDENT` instances in text corpora would be a technical task once the semantic representation was fixed. At this level, instead we face questions like, whether or not a named entity somewhere in the text detected by our text processor, e.g., "Barack Hossein", is referring to the one `PRESIDENT` instance that we named above as Obama.

Figure 1 illustrates similar levels of abstraction for event classes, event instances, and event mentions. The distinction between the second and the third layer are more obvious and previously considered as clearly in other frameworks. The distinction between the first and the second layer, though, is often left implicit, even in recently published event annotation guidelines. For example in a *Grounded Annotation for Events* (GAF, Fokkens et al. 2013), event mentions are clearly distinguished from instances. However, the first two layers have been taken as one, i.e., the *semantic layer*. In their work, *event type* which is an artifact of the adapted semantic ontology (SEM, Klyne and Carroll 2004), implicitly works similar to the classes in our definition. Nevertheless, these three layers are intuitively separable and familiar for linguists working on event and entity recognition. Bejan and Harabagiu (2010), for example, introduce the event coreference resolution with an example put into a similar three-layer hierarchy, despite their purely data-driven approach leaving off prior semantic specifications. Here, we explain each layer of the model separately. Issues specific to coreference detection will be presented in the following section.

2.1 Event Classes

The first layer of the ontology determines event type definitions. Each class can have totally different attributes depending on the interests of a particular study. Some events might be identified only by their time and place, while others by participants of prioritized importance. A very flat semantic representation would attribute all types of events with a fixed set of entities, e.g.: participants, time and location. Note, however, that structural and semantic differences exist among events of different natures, even if these complex phenomena are reduced into something more familiar and tangible such as verb frames (Fillmore et al., 2003). For example, a `KILLING` event is essentially attributed with its `Agent` and `Patient`, while salient attributes of an `EARTHQUAKE` include `Location`, `Magnitude`, `Time` and `Human Impacts`, in a typical news context. This becomes even more clear when event types are taken and compared against one another from different genres of text (Pivovarova et al., 2013; Shaw, 2013). A scientific attitude toward the analysis of `EARTHQUAKE` events might characterize them with `Natural Impacts` rather than `Human Impacts`. Thus, the first layer of the model needs to be designed with respect to the specific information extraction goals of the particular study, be it a pure linguistic or an application-oriented one.

Ambiguities about the granularity of attributes, subevent-ness, scope and most importantly, identity between event instances are dealt with at the definition layer for and between classes. For example, if the modeler wants to allow coreference between instances of `KILLING` and `SHOOTING` to indicate some type of coreference between an event and its possible subevent then this needs to be introduced at the class level, along with a procedure to compare instances of the two classes, which possess different sets of attribute[2]. Remarks

[2]The same applies even to a more flexible case, when the modeler wants to allow coreference between `KILLING` and `DYING` instances (e.g., if a `KILLING`'s `Patient` is the same as a `DYING`'s `Theme`).

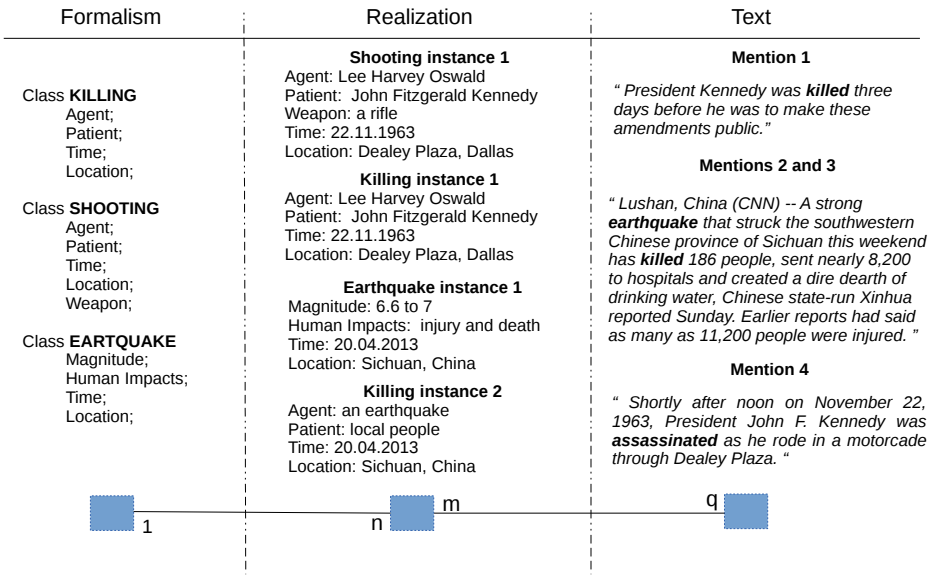

Figure 1: A three-layer ontology of events: classes, instances and mentions

of Hovy et al. (2013b) on different types of identity according to lexicosyntactic similarity, synonymy and paraphrasing indicate that the modelers have a wide choice of identity definition for event types. In section 4.3 we explain how to adapt an extended version of synonymy in order to define event classes prior to similarity-based clustering of the mentions.

2.2 Event Instances

Layer 2 indicates perfect instantiation, representative of the human common sense intuition of phenomena in real world. Instances in this layer correspond to the Davidsonian notion of events as concrete objects with certain locations in space-time, something that is happening, happened, or will happen at some point (Davidson, 1969). Therefore, links from classes to instances represent a one-to-many relation. Every instance of the EARTHQUAKE is determined with a unique set of attribute values. Two EARTHQUAKE instantiations with exactly similar attribute values are just identical. In order to keep a clear and simple representation specific to the study of coreference, the model does not allow any connection or relation between two event instances unless via their classes. Note that in Figure 1, for each realized object, only attributes included in the formalism layer are presented with their values, while in reality events occur with possibly infinite number of attributes.

2.3 Event Mentions

Facing an event mention in the text, one should first determine its class and then the unique event instance, to which the mention points. Detection of the class depends on the semantic layer definitions, while discovering the particular instance that the mention is talking about relies on the attribute values extractable from the mention context.

Usually, mentions provide only partial information about their target event instance. They can be compared against one another and (if available) against a fully representative mention, which most clearly expresses the target event by providing all necessary attribute values. Fokkens et al. (2013) refer to such a mention as the *trigger event*. Sometimes it is possible that the context is even more informative than necessary to resolve the unique real world corresponding event (see details about the impact of the earthquake in mention 3, Figure 1). In natural text a mention can refer to more than one event instance of the same type, for example when a plural case is used: *" ... droughts, floods and earthquakes cost China 421 billion yuan in 2013"*. Hovy et al. (2013b) propose partial coreference between singular and plural mentions. In

our model plural mentions are not treated semantically differently, they only point to several instances, thus, are coreferential with any single mention of them as long as the attribute values allow[3].

With respect to the above discussion, links from layer 2 to 3 represent many-to-many relations: an event instance can have several mentions in the text, and a single mention can point to more than one event instance at a time.

3 Towards Coreference Analysis

In terms of method, two different approaches have been tried in the literature under the notion of event coreference resolution (Chen and Ji, 2009; Bejan and Harabagiu, 2010; Lee et al., 2012; Hovy et al., 2013b). The first and most theoretically founded strategy is to decide for every pair of event mentions, whether or not they refer to the same event instance. Since in this approach decisions are independently made for every pair of event mentions, a clear formalism is needed to determine exactly what types of coreference are possible and how they are detected by looking at textual mentions (Chen and Ji, 2009; Hovy et al., 2013b). Some related work on predicate alignment also fit into this category of research (Roth and Frank, 2012; Wolfe et al., 2013). Alternatively, in automatic event clustering, the objective is basically discovering event instances: all we know about an event in the world is the collective information obtained from mentions referring to that in a text corpus. Each cluster in the end ideally represents a unique event in reality with all its attribute values (Bejan and Harabagiu, 2010; Lee et al., 2012). Some formal and technical differences exist between the two approaches.

Boolean choice: traditionally, clusters shape with the idea that all mentions within a cluster are of the same identity. Every randomly chosen pair of mentions are coreferent if they are found in a single cluster at the end, and non-coreferent otherwise. Therefore, taking this approach implies a level of formalism, which rules out partial coreference. On the other hand, pair-wise classification could consider partial coreference whenever two event mentions are neither identical nor totally different (Hovy et al., 2013b). Soft-clustering can compensate some deficiencies of traditional clustering approaches[4].

Transitivity: all mentions in a single cluster are coreferential, whereas pair-wise labels allow for non-transitive relations among event mentions. Depending on the specific goal of a study, this could be an advantage or a disadvantage. Lack of transitivity could be considered as an error if it is not consciously permitted in the underlying semantic formalism.

Complexity and coverage: event mentions can appear in noisy or sparse context where information for detection of their target event instance is not available. Dealing with such cases is usually easier in a clustering framework where similarity scores are calculated against the collective information obtained from a population of mentions, rather than an individual occurrence. Classification approaches could comparatively handle this only if sufficiently representative labeled data is available for training.

Exploration: a general advantage of cluster analysis is that it provides an exploratory framework to assess the nature of similar input records, and at the end it results in a global distributional representation. This is specially desired here, since computational research on event coreference is in its early ages. Evaluation corpora and methodology are still not established, thus, the problem is not yet in the phase of "look for higher precision"!

The method we are going to propose in the next section combines a rule-based initial stage with a similarity-based clustering procedure. This is partially inspired by the work of Rao et al. (2010), where entity coreference links are looked up in high-volume streaming data. They employ a lexicon of named entities for cluster nomination to reduce the search space. Once a mention is visited only the candidates among all incrementally constructed clusters up to that point are examined. Incremental clustering strategies are in general suitable for a pipeline project by efficiently providing single visits of every mention in its context. Feature values of a mention can be extracted from the document text, used for clustering, and combined

[3] The other type of quasi-identity discussed by Hovy et al. (2013b) engaged with sub-events is handled in the semantic level.

[4] For example, multi-assignment would allow plural mentions to take part in several different clusters, each representative of one event instance.

into the feature representation of the assigned cluster in a compressed format.

4 Event Coreference System

The original data in our study is a text corpus automatically annotated with several layers of syntactic and semantic information (Blessing et al., 2013). The English portion includes news and commentary articles of several British and American publishers from 1990 to 2012. An approximate average of 100 event mentions per document with the large number of total documents per month (avg. 1200) requires us to think of different ways to reduce the search space and also design a low-complexity coreference resolution algorithm.

4.1 Partitioning

In cross-document analysis, typically, a topic-based document partitioning is performed prior to the coreference chain detection (Lee et al., 2012; Cybulska and Vossen, 2013). Since we are interested to track discussions about a certain event possibly appearing in different contexts, this technique is not desired as coreference between mentions of a single real word event in two different topics would remain unknown. For example, when an articles reviews several instances of a certain event type such as different attacks that has happened in a wide temporal range and in different locations, such articles would not be included in any of the individual topics each focused on one event instance. As an alternative to the previous approach, we perform a time-window partitioning based on the article publication date before feeding the data into the coreference analysis algorithm. Larger windows would capture more coreference links: this is a parameter that can be set with respect to the available resources in trade-off with the desired search scope. In the future, we would like to invent an efficient procedure to combine the resulting clusters from consecutive time-windows in order to further enhance the recall of the system.

4.2 Event Mention and Feature Identification

In order to extract event mentions we use the ClearTK UIMA library (Ogren et al., 2008), check the PoS of the head word in the extracted text span and take all verbal and nominal mentions into account. In the current implementation all event classes are identified by a fixed set of attributes including `Timestamps` and `Related Entities`. While being very coarse-grained, this way of attribution is quite intuitive: events are identified by times, places and participants directly or vaguely attached to them. Temporal expressions are extracted also by ClearTK and normalized using SUTime (Chang and Manning, 2012). Named entities of all types except `Date` are used which are obtained from previous work on the same dataset (Blessing et al., 2013).

4.3 The Two-step Algorithm

Having all required annotations, we select a time window and perform the following two steps for event mentions of the TimeML classes `Occurrence`, `I-Action`, `Perception` and `Aspectual`[5].

1) Semantic class identification: WordNet synsets provide a rich resource in order to be adapted as event classes (Fellbaum, 1999). They cover a large lexicon and the variety of relational links between words enables us to specify a clear semantic convention for the coreference system. In addition to the mentions coming from the same synset, we allow coreference between events belonging to two different synsets that are directly connected via hypernymy or morphosemantic links. While every WordNet synset comprises words only from a single part of speech, morphosemantic relations allow the model to establish cross-PoS identity among words sharing a stem with the same meaning which is desired here: observe (verb) and observation (noun)[6]. A Java library is employed to access WordNet annotations (Finlayson, 2014).

2) Similarity-based clustering: A mention is compared against previously constructed clusters with respect to the attribute values that are extractable from its context. In order to fill the `Timestamps` attribute we have employed a back-off strategy: first we look at all time expressions in the same paragraph where the event mention appears, if we found enough temporal information, that would suffice. Otherwise, we look into the content of the entire article for temporal expressions. The `Related Entities` at-

[5]Other types, namely, `Report`, `State` and `I-State` events are not interesting for us, therefore such mentions are simply skipped.

[6]When a mention is visited all compatible synsets according to the head lemma are tried because in the current implementation we do not perform word sense disambiguation.

tribute is filled similarly by looking at the named entities in the context of the event mention. The first step is a procedure to candidate clusters containing mentions of related types. If no cluster is a candidate, a singleton cluster is created and its class is added to the index of visited event types (synsets). If candidate clusters already exist, we calculate the feature-based similarity score for each. If the best score is below a threshold a new singleton cluster is created but in this case for the reason that, perhaps, not a new type but a new event instance is visited.

5 Manual Annotation and Evaluation

The Event Coreference Bank, which is the largest available corpus with cross-document coreference labels, supports only a within topic evaluation (ECB, Bejan and Harabagiu 2010). In order to perform a more realistic evaluation of the method presented in this paper, we selected a subset of events from the AQUAINT TimeML corpus and annotated those with coreferentiality. The AQUAINT TimeML data has recently served as one of the benchmarks in the TempEval shared task (UzZaman et al., 2013) and is available for public use[7]. It contains 73 news report documents from four topics, annotated with 4431 event mentions and 652 temporal expressions which make it suitable for our task. Two main differences between our annotation and the ECB data are: 1) event mentions here are selected semi-randomly[8] and across topics rather than topic-based, 2) they are shown pair-wise to the annotator (in order to catch the transitivity patterns after the analysis), whereas, in the ECB, event mentions are clustered. Furthermore, the data already comes with manually assigned mention boundaries, event types, temporal expressions and links between events and temporal expressions, all according to the TimeML standards (Hobbs and Pustejovsky, 2003). These serve exactly as features that our algorithm uses for construction of clusters. We only had to perform named entity recognition automatically to have data ready for evaluation of the model. The manual annotation of 4950 pairs resulting from 100 selected event mentions ($\frac{100!}{2!(100-2)!}$) was done with the help of a simple user interface, which showed each of the two event mentions within its context to the annotator and asked for pushing `yes`, `no` or `next` (undecided) button to proceed to the next pair. After studying the annotation guideline published by Cybulska and Vossen (2014), our expert spent some hours during a week for the job. Decisions made in shorter than 500 ms were revised afterwards. There was one `no` answer which the annotator found unsure after revision, as it resulted in a transitivity violation, but we left it unchanged due to the nature of pair-wise decisions. In the end we came up with a total of 36 `yes`, and 4914 `no` pairs.

[7] `http://www.cs.york.ac.uk/semeval-2013/task1`

[8] Since the number of coreferential mentions is much smaller than non-coreferent ones, we adapted a heuristic measure to make sure that we will have some similar mentions among the 100 records. Therefore, we would call it a semi-random selection, still different from the fully selective strategy employed for ECB.

6 Experiments

This section provides an insight into how clusters of event mentions are created for a portion of our large news corpus. We also run the algorithm on the manually annotated data to perform an error analysis.

6.1 Construction of Event Clusters

News text from New York Times and Washington Post are combined to demonstrate a showcase of clustering for a time-window of two weeks (250 articles)[9]. Figure 2 shows the creation curve of event classes (type index entries) and event instances (clusters) as the number of the visited mentions increases. Comparison between the number of mentions with that of clusters indicates that a great deal of event instances are mentioned only once in the text. Since, for each mention, all compatible synsets are added to the type index (if not there already) during the early stages of clustering the number of the type index entries is times the number of visited mentions. In the middle to the end phases the type index contains a large collection of event classes, also a decent number of non-singleton clusters (repeatedly mentioned event instances) are created. Statistics of the type of clusters obtained after performing the algorithm on the processed mentions are presented in Table 1. A significant number of non-singleton clusters contain mentions only from a single paragraph or a single article, which is expected given the type

[9] This collection is processed within a few minutes on a normal PC by the proposed algorithm starting with zero clusters.

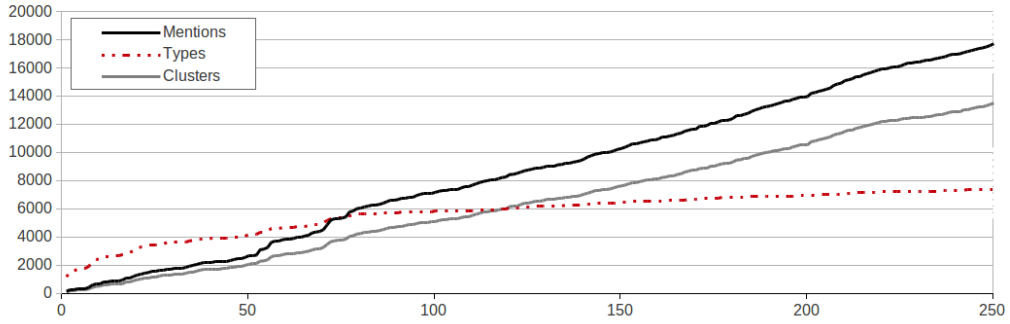

Figure 2: Number of clusters and the type index entries as mentions are visited in 250 articles

of features; remember that `Timestamps` and `Named Entities` are looked up in a paragraph scope. Clusters containing mentions from several articles, namely, the *popular* ones are most interesting for us as they would be representative of the systems performance on cross-document coreference analysis. By looking at those we found that the named entities have a very important role in finding similar subtopics within and between documents. Temporal expressions are less helpful as they are rare, and otherwise introduce some noise when documents are already being processed in a specific publication time-window. For example, the word *today* which appears in most articles of the same day (and would be normalized to that day's date, e.g., "1990.01.12") would gather mentions of a general event type, e.g., *meet*, although, they might not be pointing to the same instance. The employed semantic convention establishes a balance between efficiency and recall of the system. Nevertheless, it sometimes allows clustering of intuitively unrelated actions. In order to enhance the clustering performance in terms of the precision, we have a parameter to give priority to within synset coreference.

Cluster type	Freq.	Avg. content
Singleton	12895	1
Single paragraph	1360	2.36
Single article	807	3.95
Popular	182	2.99

Table 1: Different types of resulting clusters

6.2 Error Analysis

We fed all event mentions from the AQUAINT TimeML corpus into the algorithm exactly in the same way that we did in case of our large news corpora. The algorithm has a few parameters which we set by looking at samples of resulting clusters prior to the measurement on the labeled portion. This is a minimal NLP system given that neither syntactic/semantic dependency of entities to the event head word nor the type of attachment to temporal expressions in the context are taken into account. Nevertheless, we obtain 51.3% precision and 55.6% recall for the pair-wise coreference resolution task on the annotated data. The resulting F-score of 53.4% is comparable with the best F-scores reported in the work of Bejan and Harabagiu (52.1% on ECB for the similar task) while they use a rich linguistic feature set, as well as a more sophisticated clustering method.

Coreference	Total	Related class	Same doc.
True positive	20	100%	25%
True negative	4895	16%	2%
False positive	19	100%	36%
False negative	16	33%	7%
Total	4950	15%	2%

Table 2: Pair-wise decisions

Table 2 shows false positive and negative answers separately. As reflected in the results, positive labels are given only to mention pairs of related classes (headwords need to share a synset, or are related via hypernym and morphosemantic links in WordNet). 36% of positive labels are given to pairs within some article which is expected given that common contextual features are easy to find for them. In such cases, usually linguistic features are needed to resolve participants or the relative temporality of one mention against the other:

a. some people are **born** rich, some are **born** poor.

b. the bullet **bounced** off a cabinet and **ricocheted** into the living room.

In some cases, on the other hand, the disagreement depends on the semantic approach to the definition of identity, and therefore, is more controversial. The human annotator has apparently been more conservative to annotate coreference when the head words of the mentions were a bit different in meaning, whereas the system's decision benefited from some flexibility:

a. the immigration service **decided** the boy should go home. / they made a reasonable decision Wednesday in **ruling** that...
b. if he **goes**, he will immediately **become**...

It is not clear, for example, whether *ruling* is a subevent of the *decision* or exactly the same event. A similar distinction needs to be made in case of the false negative labels. The automatic clustering is not able to detect coreference mostly in case of sparse context, where enough information is not available to resolve the similarity. That is why false negative happens more frequently for mentions coming from different articles (specifically paragraphs sharing few named entities) and only 7% of the time when they happen within a document:

a. the Clinton administration has pushed for the boy's **return**. / his son said he didn't want to **go**.

Sparse context results either in the creation of a singleton cluster for the mention or careless assignment to some wrong cluster, which in the future would decrease the chance of meeting coreferent mentions. False negatives happening for mentions of unrelated semantic classes are due to the missing links between *possibly synonym* words in WordNet, one of the issues that need to be investigated and cured in the future work.

7 Conclusion

This paper presented a variety of material concerning event coreference resolution:

1. A general ontology is explained that can be employed in different studies on events.

2. An algorithm is designed, regardingly, to gather coreferential event in a large corpus.

3. A set of event mentions in AQUAINT TimeML is annotated with pair-wise coreference tags within and between topics[10].

4. An implementation of the method considering simple and scalable features is tested on real data and the annotated corpus.

5. Finally, we performed an error analysis of the automatically assigned labels to identify future directions.

Separating the semantic layer definition of coreference from textual attribution of event mentions has two benefits in our framework. First, it provides us with an efficient partitioning procedure to reduce the search space. Second, it makes the model flexible to allow for different possible semantic conventions which could vary from one application to another. Our adaptation of WordNet synsets allows for integrative future extension of the model — e.g., to capture metaphorical and subevent relations based on `Methonymy` and `Entailment` links. The intuition of using named entities for identification of important real-world events resulted in balanced precision and recall on the test data. In the future, we would like to investigate the effect of linguistic features on improving the performance of the algorithm. In particular, it would be interesting to see whether exact specification of event head arguments would outperform the vague attribution with related entities. The state-of-the-art result in the supervised predicate alignment approach is a hint for rich linguistic features to be helpful (Wolfe et al., 2013). On the other hand, depending on the adapted event identity definition, coreferential events might not really share identical arguments (Hasler and Orasan, 2009). There are differences between real data collections and the available annotated corpora, including ours, which needs to be investigated as well. For example, small collections do not include enough same-class event mentions pointing to different event instances, and it brings about unrealistic evaluations. Furthermore, annotation guidelines are usually biased towards a specific theory of event identity which affect the resulting data in one way or another. Some applications demand different semantic conventions perhaps with broader/narrower definition of identity. This is a dilemma that needs to be resolved through more theoretical studies in touch with real world problems such as the one we introduced in this paper.

[10]The annotation is available at: `http://www.coli.uni-saarland.de/~fatemeh/resources.htm`

References

Bejan, C. A. and Harabagiu, S. (2010). Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1412–1422. Association for Computational Linguistics.

Blessing, A., Sonntag, J., Kliche, F., Heid, U., Kuhn, J., and Stede, M. (2013). Towards a tool for interactive concept building for large scale analysis in the humanities. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 55–64, Sofia, Bulgaria. Association for Computational Linguistics.

Chang, A. X. and Manning, C. (2012). Sutime: A library for recognizing and normalizing time expressions. In *LREC*, pages 3735–3740.

Chen, Z. and Ji, H. (2009). Graph-based event coreference resolution. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 54–57. Association for Computational Linguistics.

Cybulska, A. and Vossen, P. (2013). Semantic relations between events and their time, locations and participants for event coreference resolution. In *RANLP*, volume 2013, page 8.

Cybulska, A. and Vossen, P. (2014). Guidelines for ecb+ annotation of events and their coreference. Technical report, Technical Report NWR-2014-1, VU University Amsterdam.

Davidson, D. (1969). The individuation of events. In *Essays in honor of Carl G. Hempel*, pages 216–234. Springer.

Fellbaum, C. (1999). *WordNet*. Wiley Online Library.

Fillmore, C. J., Johnson, C. R., and Petruck, M. R. (2003). Background to framenet. *International journal of lexicography*, 16(3):235–250.

Finlayson, M. A. (2014). Java libraries for accessing the princeton wordnet: Comparison and evaluation. In *Proceedings of the 7th Global Wordnet Conference*, pages 78–85.

Fokkens, A., van Erp, M., Vossen, P., Tonelli, S., van Hage, W. R., SynerScope, B., Serafini, L., Sprugnoli, R., and Hoeksema, J. (2013). Gaf: A grounded annotation framework for events. In *NAACL HLT*, volume 2013, page 11.

Hasler, L. and Orasan, C. (2009). Do coreferential arguments make event mentions coreferential. In *Proc. the 7th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC 2009)*.

Hobbs, J. and Pustejovsky, J. (2003). Annotating and reasoning about time and events. In *Proceedings of AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*.

Hovy, E., Mitamura, T., and Palmer, M. (2013a). The 1st workshop on events: Definition, detection, coreference, and representation.

Hovy, E., Mitamura, T., Verdejo, F., Araki, J., and Philpot, A. (2013b). Events are not simple: Identity, non-identity, and quasi-identity. *NAACL HLT 2013*, page 21.

Klyne, G. and Carroll, J. J. (2004). Resource description framework (rdf): Concepts and abstract syntax. w3c recommendation, 10 feb. 2004.

Lee, H., Recasens, M., Chang, A., Surdeanu, M., and Jurafsky, D. (2012). Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500. Association for Computational Linguistics.

Ogren, P. V., Wetzler, P. G., and Bethard, S. J. (2008). Cleartk: A uima toolkit for statistical natural language processing. *Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP*, 32.

Pivovarova, L., Huttunen, S., and Yangarber, R. (2013). Event representation across genre. *NAACL HLT 2013*, page 29.

Rao, D., McNamee, P., and Dredze, M. (2010). Streaming cross document entity coreference resolution. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1050–1058. Association for Computational Linguistics.

Roth, M. and Frank, A. (2012). Aligning predicate argument structures in monolingual comparable texts: A new corpus for a new task. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared*

task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation, pages 218–227. Association for Computational Linguistics.

Shaw, R. (2013). A semantic tool for historical events. *NAACL HLT 2013*, page 38.

UzZaman, N., Llorens, H., Derczynski, L., Verhagen, M., Allen, J., and Pustejovsky, J. (2013). Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second joint conference on lexical and computational semantics (* SEM)*, volume 2, pages 1–9.

Wolfe, T., Van Durme, B., Dredze, M., Andrews, N., Beller, C., Callison-Burch, C., DeYoung, J., Snyder, J., Weese, J., Xu, T., et al. (2013). Parma: A predicate argument aligner.