CS 224 U: Literature review

Aju Thalappillil Scaria, Rishita Anubhai, Rose Marie Philip
ajuts, rishita, rosep

Task definition

The web has an enormous amount of information stored as text content. While this information is accessible to anyone with an internet connection, analyzing this vast amount of data to extract meaningful information is rather difficult because of its sheer volume. Building automated mechanisms to parse the content of the web to build automatic question answering systems has been an area of reasearch for many years. For instance, extracting events and the associated entities from biomedical and molecular biology text has drawn a lot of attention, especially because of the introduction of the BIONLP tasks targeting fine-grained information extraction. This has also been motivated by the increasing number of electronically available publications stored in databases such as PubMed. The task of event and entity extraction encompasses several areas including natural language processing, computational linguistics and text mining. This task would in turn help other popular NLP tasks such as question answering and machine translation.Different methods that make use of syntactic and dependency parsing, semantic role labeling, machine learning techniques etc. have been used over the years to tackle this challenging problem. While some of the systems built have had significant improvement over the previously existing ones, the performance of the state-of-the-art systems is still quite distant to that of a 'perfect' extraction system. This is mostly because interpreting meaning from complex structures of natural language is a hard task. In this paper, we review nine research publications from the area of event and entity extractions. In the first section, we summarize the content of the literature reviewed and highlight their main design characteristics and contributions. Then, we compare the different approaches taken in the papers to compare and contrast the different strategies that have adopted, their similarities and differences.

Summaries of articles

1. Unsupervised Learning of Narrative Event Chains - Chambers and Jurafsky (2008).

This paper talks of just the areas of event extraction and temporal event ordering. It does argument extraction at a very crude level. This paper tried to work around the need for highly supervised scripts, by trying to generate event chains that they call narrative event chains in an unsupervised manner. They follow the intuition that events related to a common protagonist, the common argument for the event verbs, should form a valid narrative event chain. This intuition builds up on the idea of case-frames and anchor based pairwise event relations models. In this paper (unlike the second and third) they do not pay much attention to the protagonist or the type of the argument as such. This paper also constructs a novel scoring method that involves two kinds of scores:

- A score based on PMI that is give to events, pairwise, depending on how often the two events share grammatical arguments.
- A score for the global narrative chain where all events in the chain provide some input (example, PMI) with a target event in question. This could be used to predict the next most likely event.

Moreover, an interesting idea that emerges out of this paper is that of testing event extraction models. The novel approach here is the narrative cloze testing model, where you leave out a particular event in the chain and compare it to the event that the model you trained would predict. A temporal classifier attends to the second of the areas listed above. The paper just explores the establishment of the before relationship in case of Temporal order identification. A major result that emerges from the paper is that the protagonist approach obviates the need for a presorted topic list of documents to perform event extraction and inferences.

2. Unsupervised Learning of Narrative Schemas and their Participants - Chambers and Jurafsky (2009).

This paper addresses narrative event chains in further detail. In the broader perspective, it addresses the areas of Event identification and Argument Extraction listed above. Event identification is extended to identify not only single event chains but to find overlapping chains and form, what is defined as a narrative schema now. Argument extraction is also done as unsupervised semantic role labeling which obviates the need of a predefined class of roles or hand built domain knowledge. These two endeavors also contribute to the improvement of each other since, as suggested above, rich event extraction largely revolves around extracting correlated events that in turn depends on finding coreferring arguments and linking them. If narrative chains are formed only based on the frequency with which two verbs share their arguments, ignoring the features of the ar-

guments themselves, then it is likely that word sense ambiguities cause the wrong event to be clustered with other events. Instead, having attached types for the arguments and modeling argument overlaps across all pairs is what this paper advocates. The semantic role labeling task that the paper takes up, is done in an unsupervised manner and by learning the roles automatically rather than picking them from a pool of predefined domain of roles. Such narrative event chains are called typed narrative chains. The types are picked by finding the most frequent head-word in co-referential chains for the arguments. These narrative typed chains are combined into narrative schemas by scores depicting chain similarities and narrative similarities. These schemas are directly comparable to frames. The major difference is that schemas focus on events in a narrative and frames revolve around certain participants.

The higher-level outcomes from the paper are that semantic role labeling and event extraction and chaining, can be put together. This could be done by unsupervised learning and lead to an improvement in the results for both of these NLP goals individually.

3. Template-Based Information Extraction without the Templates - Chambers and Jurafsky (2011).

This paper deals with information extraction for similar purposes as event extraction, but without using predefined templates for the same. The major novelty of the paper lies in the method introduced in the paper to learn the template structure from an unlabeled corpus. This work runs in parallel with relation discovery, frames, scripts and narrative schemas for the purpose of information extraction. The main positive of this paper over other approaches is that redundant documents about specific events are not needed. Also, templates with any type and number of slots can be filled in an unsupervised fashion.

The main aim of the paper is to extract information from a domain specific corpus by learning a richly understood template structure. The method for this involves

- Expanding the corpus size by assembling a larger information retrieval corpus of documents for each cluster.
- Clustering the words in the events based on their proximity, coreferring arguments and selectional preferences.
- Inducing semantic role labels.

The clustering of event patterns is done by using LDA and agglomerative clustering based on word distance, PMI over all event patterns. Chambers and Jurafsky (2009) recommended methods to learn situation-specific roles over narrative schemas. This paper introduces a novel vector approach to co-reference similarity. This approach builds on the intuition in **?** and Chambers and Jurafsky (2009) that coreferring arguments imply a semantic relationship or event chaining between two predicates. This idea is applied to build the vector similarity framework to perform role labeling. Document classification is analyzed as an example of information extraction based on these concepts then.

The paper talks about over all template evaluation as well as a stricter per template evaluation. The results show that the MUC-4 template structure was learnt with many new semantic roles and template structures. Moreover, the results have comparable precision and an F1 score that approaches existing algorithms which rely heavily of prior knowledge of the domain.

Since this approach is unsupervised, the recall is hurt. Also since event extraction here happens as an after stage of template induction, the number of parameters required are high and could be reduced in future work.

4. Extracting Complex Biological Events with Rich Graph-Based Feature Sets - Bjorne et al.(2009).

This paper describes a system for extracting events among genes and protiens from biomedical literature. Their model handles the situations in which an event is an argument to other events, resulting in a nested structure. They divide the task of event extraction into three independent steps considered as machine learning problem making use of features using dependency parse graph - trigger recognition, argument detection and semantic post processing. They claim that by separating trigger recognition from argument detection, they can use methods like named entity recognition to tag words as entities. But, they assume that all entities are recognizable as named entities and are availalbe already. In their model, the sentence or paragraph from which events are to be extracted is represented as a graph with nodes corresponding to entities (or arguments) and events, and the edges corresponding to event arguments. This is a natural representation as it is easy to encode event-event relations as well. As the first step, they tackle the problem of trigger detection as a token labeling problem in which each token is assignned to an event class if it is likely to be an event otherwise, to the negative class. Multi-class SVM is used for the classification task. Token features (related to properties of individual tokens like capitalization, presence of punctuation etc), frequency features (for e.g., number of entities in the sentence), and dependency chain features (encoding dependencies between tokens) of length upto three are used. In the next stage, they have edge detection, modeled again as a multi-class SVM to tag each potential edge between an event and an entity; or an event and an event representing a relation between them. The edge is tagged as *theme, cause* or a negative denoting the absense of an edge. The features they use include N-grams (generated by merging attributes of 2-4 consecutive tokens), individual component features, semantic node features (obtained from just the two nodes) and frequency features. In this case, the edges between different nodes (entities and events) are predicted independent of each other. Because of this, they need a third stage in their pipleline to do semantic post processing to ensure that the semantic graph produced by the trigger and edge detection steps does not have any improper combinations. This is a rule based step which refines the graph based on event-event and event-argument types. They also do steps like node duplication in case there are two separate events that are denoted by a single word.

5. Joint Learning Improves Semantic Role Labeling - Toutanova et al.(2005). They use a joint model to improve semantic role labeling. In their support, they claim that there are tight dependencies between the different entities and their arguments, because of which, building independent classifiers to handle the task of event extraction as-

sumes a lot of independence which does not really exist in natural language. For instance, there may be hard constraints according to which arguments to events (or predicates) cannot overlap with each other, and also soft constraints by which a predicate cannot have two or more agents as arguments. In this paper, they present a joint model for semantic role labeling using global features to achieve better performance as compared to the state of the art models. In their model, they had two sets of models - local models that learn to role label nodes in the parse tree independently, called as *local models* and models that incorporate dependencies among the labels of multiple nodes, called *joint models*. The local classifier handles the task of semantic role labeling as two separate tasks of identification and classification. The features that they use are tightly bound to the parse tree and parts of speech tags. In the identification phase, each node of the parse tree created from the sentence is tagged as an argument or not, and, in the next stage, each argument is associated with its appropriate semantic role. This is possible because these tasks of identification and classification decomposes well, so that they can be solved separately. Since the local models don't capture the dependencies amonst arguments, they also have a joint model which builds on top of assignments generated by the local model. Their model also does re-ranking of the assignments generated by the local model based on weights learnt from a log-linear re-ranking model. For this, they map the features from a parse tree and the label sequence to a vector space and then maximize the log likelihodd of the best assignments. One important factor that helps their model perform well is the usage of templates to generate features. A template helps capture dependencies between label of a node and input features of other argument nodes - for instance, templates could be used to avoid picking multiple arguments to fill the same semantic role (agent of the verb for instance) or, to count the number of arguments ro the left and right of the predicate. This model gave them better accuracies than the best model. In short, jointly modeling the arguments of verbs does help.

6. Fast and Robust Joint Models for Biomedical Event Extraction - Riedel and McCallum (2008). introduce three joing models of increasing complexity designed to extract bio-medical events. They formulate the search for event structures as an optimizaton problem through a set of binary variables by projecting events to a graph structure over tokens. For a sentence and a set of candidate trigger token, they label each candidate with an event type, t it is trigger for, or 'None' if it is not a trigger. Hence, for a candidate trigger, there are as many binary variables as the nubmer of possible event types + 1. Let these variables be denoted by e. For each candidate trigger, they introduce binary variables a to asscociate the trigger to all arguments a of the event to which the trigger belongs to. These arguments can be events or entities. This variable are used to add association between event-event and event-entity pairs. This representation has the shortcoming that it is not possible to differentiate between two events with the same trigger but different arguments, or, one event with several argument. While earlier work overcame this by adding adhoc rules, Riedel and McCallum (2008) augmented the graph representation by adding edges between the pair of arguments that are a part of the same event by introducing another binary variable b. As mentioned before they have three progressive models for event extraction. The first model does joint trigger and argument extraction (by learning the assignment variables e and a) by an efficient exact inference algorithm by independently scoring trigger labels and argument roles and maximizing the sum of the scores both by using a scoring function. Model 1 can predict structures that cannot be mapped to events. For instance, it can label a token that is actually an event as an argument to another event, but, the former event may not be an active trigger or argument. This would not be ideal as we would want events to combine only with events or arguments. Model 2 enforces these constraints by using a scoring function to identify consistent trigger labels and incoming edges. Model 3 predicts the variables that denote argument-argument combinations as described earlier by using a scorign function to learn weights for the binary variable b. In short, their work model the full process of event extraction as an optimization problem over a set of binary variables.

7. Event Extraction as Dependency Parsing - McCloskly et al.(2011). introduce the task of event extraction from text by means of dependency parsing. Nested event structures are common occurrences, but most model before them were incapable of handling them as events and arguments were extracted independently. In this paper, they propose extracting events (including nested events) by taking the tree of event-argument relations and using it directly as a representation in a reranking dependency parser. The entities in this task were a part of the dataset and were provided, but the event anchors were predicted by a multi-class SVM classifier. In the first phase, the original event representation is converted to dependency trees containing event anchors and entity mentions. Each event anchor is linked to each of its arguments and is labeled with the slot name of the argument. The labeled dependency links were generated using MSTParser. In this model, the graph may contain self-referential edges due to related events sharing the same anchor. Since their re-ranking algorithm would work only on trees, they had several preprocessing steps. They removed self-referential edges and also broke structures where one argument participates in multiple events by keeping only the dependency to the event that appears first in text. Also, all events with same types anchored on the same anchor phrase was unified. After this phase, the resulting dependency structure is int he form of a tree. The MSTParser used features that were edge factored - that is, the features are extracted based on the features of end points of the edge and that of the edge itself. Everytime the MSTParser was run, it finds the highest scoring tree that incorporates the global properties that included event path (path from each node in the event tree upto the root), event frames (event anchors with all their arguments and argument slot names). Since their approach is not restricted by sentence boundaries, it could be extended to work on entire documents.

8. Jointly Combining Implicit Constraints Improves Temporal Ordering - Chambers and Jurafsky (2008). This paper discusses ways to improve existing work on or-

dering events in text. Previously, event-event ordering tasks were based on local pairwise decisions using classifiers like SVM using different features of the event like text of event, WordNet synset, POS tags, tense, etc. But these could sometimes introduce global inconsistencies when misclassifications occur, that are plainly obvious. For example, if event A occurs before B, B occurs before C, then A cannot occur after C. This paper tries to repair some of these event ordering mistakes by introducing two types of global constraints: transitivity and time expression normalization. In short, this work is on classifying relations between events, making use of relations between events and times, and between the times themselves.

The dataset had newswire articles that are hand-tagged for events, time expressions and relations between events and times. The events are also tagged for temporal information like tense, modality, grammatical aspect etc. The first model for event-event ordering (to *before and* after) uses a pairwise classifier between events and a global constraint satisfaction layer (using an integer linear programming framework) that re-classifies certain examples from the first stage if it seems to violate properties like transitivity. These constraints can also help create more densely connected network of events by adding implicit relations that are not labeled. For example, if A occurs before B and B occurs before C, we can add the relation A occurs before C. However it was seen that having this additional layer did not change the results globally because the hand-tagged data had large amount of unlabeled relations and global constraints cannot assist local decisions if the graph is not connected.

So, an addition was made to the model - time-time information that are deduced from logical time intervals. For example, if event A occured 'last month' and B occured 'yesterday', we can conclude that A occured before B because 'last month' occured before 'yesterday'. Having this additional feature along with the global constraints, greatly increased the size of training data set (81% increase) and also improved the performance (3.6% absolute over pairwise decisions).

Thus this paper shows how textual events can be indirectly connected through a time normalization algorithm that creates new relations between time expressions and that this increased connectivity is essential for a global model to improve performance.

9. Joint Inference for Event Timeline Construction - McCloskly et al.(2012). This paper tries to map events into a timeline representation where each event is associated with a specific absolute time interval of occurence rather than just inferring the relative temporal relations among the events.

The data is simliar to Chambers and Jurafsky (2008) where events in news articles and associated arguments are hand labeled. Four relations between events are considered - *before, after, overlap and no relation*. A time interval is represented in the form [start time, end time]. These intervals are sometimes explicitly mentioned in the text while at other times, it might have to be inferred relative to the document creation time of the article.

The model has three steps: 1) two local pair wise classifiers, one between event mentions and time intervals (E-T) and another between event-event mentions(E-E) 2) a combination step with event coreference (discussed later) to overwrite prediction probabilities in step 1 and 3) a joint inference module that enforces global cohernecy constraints on the final output of the local classifiers.

Classification in the local level is done using regularized average Perceptron over all possible pairs of event/time mentions using several features like the word, lemma, POS of events mentions, position of entities, tense, type of time interval, etc. Event coreference information is used to enhance the timeline construction performance, because all mentions of a single event overlap with each other and are associated with the same time interval. Also, all mentions of an event have same temporal relation with all mentions of another event. These two properties help avoiding misclassification in a lot of cases. The global inference model combines the local pairwise classifiers through the use of an Integer Linear Programming formulation of constraints. Both E-E and E-T tasks are optimized simultaneously.

The results showed that event coreference improved the performance of the classifier. The performance is better than most of the reported models and having time intervals instead of time points lowers the running time of the algorithm considerably.

Compare and contrast

In our literature review, we cover research papers that broadly fall into two categories. One category deals with event extraction and semantic role labeling. The other deals with temporal ordering of events. We also looked at literature that covered biological event extraction for the BIONLP task for gaining further insights into event extraction specific to a popular domain.

The three paper series that we chose(Chambers and Jurafsky (2008), Chambers and Jurafsky (2009), and Chambers and Jurafsky (2011)), builds gradually on the different aspects of event extraction. The motivation behind event extraction of this kind can often be NLP applications such as question answering and machine translations, since event models make this easy. Broadly, event extraction, as explained by Chambers and Jurafsky over the three papers comprises of the following main areas:

1. Event identification
2. Temporal order identification
3. Argument Extraction (which may further become semantic role labeling)

Thus, the three paper series, broadly deals with one or more sections of these while discussing event extraction.

Chambers and Jurafsky (2008) discusses chaining of events based on just one argument or participant. The second paper in the series, Chambers and Jurafsky (2009) builds up on that by representing other entities involved in the events as well. This information of recognizing multiple entities can be very valuable, since now it is possible to overlap various event chains and form a larger event scenario or a 'narrative event schema' as described in the second paper Chambers and Jurafsky (2009). Chambers and Jurafsky

(2008) also does not pay much attention to the type or role of the protagonist. Role information such as knowing if the protagonist is a place or a person or an object could help in clustering the events as well. More than mere verb comparison based on exact same argument, role and type information given to arguments in Chambers and Jurafsky (2009) leads to a stronger approach for understanding the event chains and finding the next most likely event. Moreover, another NLU task of semantic role labeling is also solved and linked to the task of event extraction herein.

In the third paper in the series, Chambers and Jurafsky (2011), event extraction here is done one step after template induction. Co-reference similarity vector framework is a novelty for clustering events. This is unlike simplistic protagonist like approaches to forming event chains. Although this is a very systematic approach for information extraction without specific requirements, it might be overkill to use a learning approach with so many parameters and template induction for just event extraction. The work by Bjorne et al.(2009) performs event extraction by a pipeline of three independent steps of identifying triggers first, followed by argument extraction and then semantic prost processing. This method is prone to cascading errors introdcued in early stages of the pipeline. For instance, if a trigger is missed in the first stage, we will never be able to extract the full event that it results in. Even though this can be tackled by passing several additional candidate to the next stage, this will increase the false positive rate as highlighted by (Miwa et al. 2010c). In addition, these models cannot make use of the rich set of features by looking at how different events and entities interact with each other. In addition, the different rule based post-processing steps that need to be used to clean up the event-entity combinations extracted might lead to partially correct relations to be thrown out because of errors introduced in some stage of the pipeline. In short, joint models helps to capture the dependencies better and are more robust.

Both Bjorne et al.(2009) and Riedel and McCallum (2008) use graph structures to encode the event-entity and event-event relationships, although in different ways. In Bjorne et al.(2009) event extraction is done disjointly and NER nodes (known) and nodes for triggers(predicted by a classifier) are joined by edges. Riedel and McCallum (2008) uses graphs to generate binary variables as edges denoting entity-entity relationships. But, they solve the problem of event extraction as an optimization problem over binary variables.

Also, Bjorne et al.(2009), Chambers and Jurafsky (2008) and Riedel and McCallum (2008) have the entities taking part in events already provided to them and focuses on event extraction, while Toutanova et al.(2005) concerns with semantic role labeling and models argument prediction jointly with event prediction task by building an ensemble of local and global classifiers.

Riedel and McCallum (2008) uses the parse tree structure quite extensively for the task of semantic role labeling. Whereas the other models for event extraction doesn't use much of syntactic information. For instance, the re-ranking model in Riedel and McCallum (2008) maps the features from the parse tree directly into vector space. This helps them use more features relating to ordering and hierarchy of words. Chambers and Jurafsky (2008) classifies the temporal relations between two events with an SVM model using local features pertaining to the events themselves. Chambers and Jurafsky (2008) also uses this kind of model and stick to classifying only the before/after relation between events, but the latter also tries to overcome some of the global problems associated with localized event ordering algorithms. However they classify the events, without identifying association rules between events and their absolute time of occurence like the work of McCloskly et al.(2012).

The three papers add up gradually on work in the area of classification of temporal relations between two events. The timeline representation for ordering events has some advantages over temporal graph representations in the former papers. The timeline model allows events to be associated with precise time intervals, which improves human interpretability of the temporal relations between events and time. It also simplifies global inference formulations as the number of variables and constraints needed in the ILP is more concise relative to time point based formulation.

Chambers and Jurafsky (2008) assumes the event-time relation to be given and tries to improve event-event temporal relation classification, whereas the model of McCloskly et al.(2012) jointly optimizes both tasks at the same time. For local classifiers, it can be seen that all three papers use similar types of features and the latter two encode transitive closure of relations between event mentions within the global inference model. Also, the two later models use Integer Linear Programming for performing the joint inference with a set of global constraints that enforce global coherency and this is seen to be better than the greedy strategy of adding pairs of events one at a time, ordered by their confidence. The comparison of classifiers also shows that McCloskly et al.(2012) could improve the accuracy of Chambers and Jurafsky (2008) to more than 5

Future work

In this paper, we investigated in detail about the different models that have been designed for event and argument extraction. Even though a lot of work has been done in this field, we feel there is a lack of focus on extending these models to build automatic question answering systems based on the events extracted. For instance, there is no system,whyen given a textbook in biology as input, extracts different high level processes involved and is capable of answering questions that involve the different events, its prerequisites, entities involved, output, and its temporal relation to other events. Developing such a system would find application in a multitude of areas in this age of information explosion. To start with, it could help students to look for information in text books or encyclopedias without having to go through them page by page. There is also a huge amount of data available in the web, but unless we have an efficient event extraction system, most of it will lie unusable. Of course, we cannot rely fully on the data available from the web, but again, we can use different heuristics to build confidence on

the data extracted as we see more supporting evidence, for instance.

References

Nathanael Chambers and Dan Jurafsky 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08*.

Nathanael Chambers and Dan Jurafsky 2009. Unsupervised Learning of Narrative Schemas and their Participants In *Proceedings of ACL-09*.

Nathanael Chambers and Dan Jurafsky 2011. Template-Based Information Extraction without the Templates In *Proceedings of ACL-11*.

Jari Bjorne, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkala, Tapio Salakoski 2009. Extracting Complex Biological Events with Rich Graph-Based Feature Sets. In *Proceedings of the Workshop on BioNLP: Shared Task,*.

Kristina Toutanova, Aria Haghighi, Christopher D. Manning 2005. Joint Learning Improves Semantic Role Labeling. In *Proceedings of ACL 2005*.

Sebastian Riedel Andrew McCallum 2008. Fast and Robust Joint Models for Biomedical Event Extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

Makoto Miwa, Rune Saetre, Jin-Dong D. Kim, and Junichi Tsujii 2010c. Event extraction with complex event classification using rich features In *Journal of bioinformatics and computational biology*.

David McClosky, Mihai Surdeanu, and Christopher D. Manning 2011. Event Extraction as Dependency Parsing In *Proceedings of ACL-11*.

Nathanael Chambers and Dan Jurafsky 2008. Jointly Combining Implicit Constraints Improves Temporal Ordering In *EMNLP-08*.

Quang Xuan Do, Wei Lu, Dan Roth 2012. Joint Inference for Event Timeline Construction In *EMNLP-12*.