ISSN 2181-3187

BASIC TERMS FOR SENTENCE-LEVEL EXTRACTION

Lutfieva Amina Safoiddinovna
Master student of Samarkand state institute of foreign languages

Annotation

In this article, we treat event detection as a sentence level text classification problem. Moreover, we clarified some basic terms in sentence–level event extraction. Our results show that the most effective classification approach is dependent on the target event type. We also investigate a rule-based method. Overall, we compare the performance of discriminative versus generative approaches to this task.

Key words: *event extraction, document-level extraction, An event trigger, An event mention, An event argument.*

The task of sentence-level event extraction differs from document-level extraction primarily along the axis of granularity. While document-level extraction assumes the presence of a single primary event, sentence-level extraction can have an unbounded number of events present within the text, each one associated with any number of event arguments. As a consequence of this, there is no notion of a “primary” event in sentence-level extraction. A side-effect of this is that human analysts are unable to determine the importance of different events without reading through the original text themselves.

The earliest event extraction system was FRUMP. The goal of FRUMP was to skim input news articles and extract events describing the most important aspects of the text. FRUMP achieved this by using two components – a predictor and a substantiator – to collaboratively fill up an event template. The predictor would be in charge of predicting what event frames exist in the text, while the substantiator would find evidence to fill up frames suggested by the predictor. Based on the evidence received from the substantiator, the predictor could then narrow down the set of possible candidate event templates, and issue new requests to the substantiator to fill up remaining slots in the template.

Historically, the task of sentence-level event extraction originates with the ACE program. A key difference with the types of events that have been studied at the sentence-level compared to document-level research is the generalizability of event types. Event types studied at the sentence-level have focused on more general themes, such as conflicts, transportation of people/items, and life events. This has allowed sentence-level work to capture a much wider range of event types than has been seen under MUC-centric document-level analysis.

ISSN 2181-3187

Let us begin by addressing common terminology seen in the literature for sentence-level event extraction:

• *An event* is something that happens in the world at a particular place and time.

• *An event mention* is a particular occurrence of an event in a document. An event may be mentioned multiple times within the same document, or the same event may be mentioned across a set of documents.

• *An event trigger* is a particular word or phrase that signifies the existence of an event.

• *An event argument* is an entity that fulfills some role within a particular event. The set of valid roles for an event depends on the type of event, including roles such as Agent, Place, and Time.

• *An event argument mention* is a particular textual instance of an event argument.

The classic approach to sentence-level event extraction is to break the problem down into a pipeline of individual subtasks – namely, trigger identification, trigger classification, argument identification, and argument classification. We describe each of these tasks below:

• Trigger identification – for every word in the document, the system must make a binary prediction as to whether or not the word triggers an event (of any type).

• Trigger classification – given the words that have been identified as event triggers, classify each of them into specific event types (e.g. Attack, Demonstration, etc.)

• Argument identification – given a set of candidate entity mentions (for example, obtained from Named Entity Recognition) and the set of classified event triggers, identify which entity mentions are associated with which events.

• Argument classification – given the set of entities associated with each event trigger, classify the relationship within each (entity, event trigger) pair into a specific argument type (e.g. Buyer, Seller, Attacker, Place, Time, etc.)

In some approaches, identification and classification steps are merged into a single classifier, resulting in a two-stage pipeline instead. Notably, almost all methods for sentence-level event extraction utilize machine learning methodologies. While early systems utilize some pattern matching for trigger predictions, the vast majority of systems rely solely on machine learning techniques for classification. This is a vast departure from document-level event extraction, where nearly all early methods relied on handcrafted rules or pattern-matching approaches, with classification only becoming popular in later years.

Instead each sentence forms a training/test instance for our classifier and is encoded using the following set of features:

ISSN 2181-3187

– Terms: Stemmed terms (using Porter's stemming algorithm) with a frequency in the training data greater than two, were used as term features. All stop words were removed from this feature set.

– Lexical Information: The presence or absence of each part of speech (POS) tag and chunk tag was used as a feature. We used the Maximum Entropy POS tagger and chunker. The POS tagger uses the standard set of grammatical categories from the Penn Treebank and the chunker recognises the standard set of grammatical chunk tags: NP (Noun Phrase), VP (Verb Phrase), PP (Prepositional Phrase), ADJP (Adjective Phrase), ADVP (Adverb Phrase) and so on. Chunk tags are used widely within the Computational Linguistics community to represent phrasal-level clauses in a span of text. For example, if a sentence contains any noun phrase, its corresponding NP chunk feature would be assigned the value '1'. Otherwise, if no noun phrase were present, the value assigned to this feature would be '0'.

– Noun Chunks: Noun chunks with a frequency greater than two were also used as a feature. Examples include 'American soldier' and 'suicide bomb'. – Additional: We added the following additional features to the feature vector: sentence length, sentence position, presence/absence of negative terms (e.g., no, not, didn't, don't, isn't, hasn't), presence/absence of a modal terms (e.g., may, might, shall, should, must, will) and the presence/absence of a location, person, organisation and a timestamp. Time-stamps were identified using in-house software developed by members of the Language Technology research group at the University Melbourne. Our belief is that these additional features will aid the learner to correctly identify onevent sentences of the target event. For example, intuitively sentences at the beginning of a document are more likely to be on-event sentences since the lead sentences of a document are often used to describe the major events discussed in the article. Therefore, we expect that the 'sentence position' feature will prove useful for this task.

References

1. **Martina Naughton, Nicole Stokes. 2010, April. Sentence-level event classification in unstructured texts. Information Retrieval Journal.**
2. Ahn, David. 2006. The stages of event extraction. In Proceedings of the ACL Workshop on Annotating and Reasoning about Time and Events, Pp 1–8, Sydney, Australia, July.
3. Allan, James, Jaime Carbonell, George Doddington, Jonathon Yamron, and Yiming Yang. 1998. Topic detection and tracking pilot study. final report.
4. Cohen, Jacob. 1960. A coeficient of agreement for nominal scales. Educational and Psychological Measurement, 20(1): Pp37–46.
5. Curran, James, Stephen Clark, and Johan Bos. 2007. Linguistically motivated large-scale nlp with c & c and boxer. In Proceedings of the ACL 2007 Demonstrations Session (ACL-07 demo), pp 29–32.

ISSN 2181-3187

6. Filatova, Elena and Vasileios Hatzivassiloglou. 2004. Eventbased extractive summarization. In In Proceedings of ACL Workshop on Summarization, pp 104 – 111.
7. Joachims, Thorsten. 1998. Text categorization with support vector machines: learning with many relevant features. In Nedellec, Claire and C ´ eline Rouveirol, editors, ´ Proceedings of the 10th ECML, pages 137–142,
8. Chemnitz, DE. Springer Verlag, Heidelberg, DE. Kraaij, Wessel and Martijn Spitters. 2003. Language models for topic tracking.
9. Bekhbudieva P.Sh and Lutfieva A.S “Sentence-level event classification in unstructured contexts” newjournal.org. 2022