Contextual Event Based News Aggregation

Atul Sehgal : asehgal@usc.edu
Kalhan Koul : kkoul@usc.edu
Pulkit Manocha : pmanocha@usc.edu
Puneet Koul : puneetko@usc.edu

June 17, 2017

1 Introduction

Event Based News Aggregation is a framework that achieves the following objectives:

- Recognizing and extracting "events" phrases from a given news article in Spanish. Event detection and extraction is a major challenge in academia right now with most contemporary research focused on an extremely broad definition of the term. We build a system that can distinguish ordinary verbs from highly-relevant events in a given text.
- Retrieving relevant information from several news sources pertaining to the above identified events and generating a condensed, cogent summary from several news sources that captures the essence of those events.

Considering the massive volume of data produced, it has become imperative to reliably extract useful and relevant information in condensed form. For example, someone reading a news article on the decline of democracy in Turkey might not be informed on the event references to the riots, electoral rigging and political events referenced or mentioned in the article. Our framework detects and extracts these event phrases for the reader and fetches the relevant information from across various news sources pertaining to them.

We define the event-phrase to be: "A non-trivial combination of entities that denotes a temporal occurrence in the real world".

This problem is exciting for several reasons. First, there is little research in NLP that concretely tackles this precise problem. Most state of the art work deals with annotating verb phrases and categorizing them as various kind of events based on grammatical usage(Yang & Mitchell, 2016). This isn't helpful when parsing a big article where only few specific events mentioned are relevant and important. Our contribution is a framework that employs more sophisticated techniques to extract highly relevant event terms from an article, builds (using semantic role labeling) event-phrases which are used for querying the web and aggregating information from different sources which is then presented as a summary. Major challenges included semantic role labeling, generation of linguistically sound event phrases, getting only those events that merit investigation and generating a summary from information retrieved.

2 Materials

For event detection and extraction, we used the following approaches and corresponding material.

- A rule based dependency parser: We used the google NLP API (Google, n.d.)to identify entities in text and used rules to build semantic relationships between them, and therefore did not require any training material for this task.

- Perceptron and Neural Networks: Training neural networks to understand linguistic structure of events and recognize event phrases from news articles, we primarily used two sources:

  – TempEval-2 training data(TempEval-2, n.d.) which was provided for the 2010 and 2012 SemEval competitions organized by ACL SIGLEX (an umbrella organization for researchers in computational lexical analysis and semantic evaluation (SemEval). This data consists of 47000 tagged Spanish tokens annotated using the TimeML markup scheme (adopted as the standard in NLP community for event annotation)

  – Manually tagged news articles [CNN and BBC]: We hand-tagged 1950 lines/sentences (from over 100 articles ranging from politics to sports). We annotated events with tags such as:

```
<root>
    <article>
        <text>
            The <e1> assassination of kennedy</e1> in
            1963 has come under investigation.
        </text>
        <summary>
            .........
        </summary>
    </article>
</root>
```

  Furthermore, we used the Stanford NLTK library(Stanford, n.d.) to annotate our corpus with POS tags.

- For training TF-IDF corpus we used the data provided by Mark Davies, Professor of Linguistics at Brigham Young University (University, n.d.) comprising over 1.8 billion spanish tokens.

3 Procedure

We split our framework in two major sections (Detecting events and generating summary of information retrieved about those events from the web):

- Detecting and extracting events. For this we tested 3 major approaches:

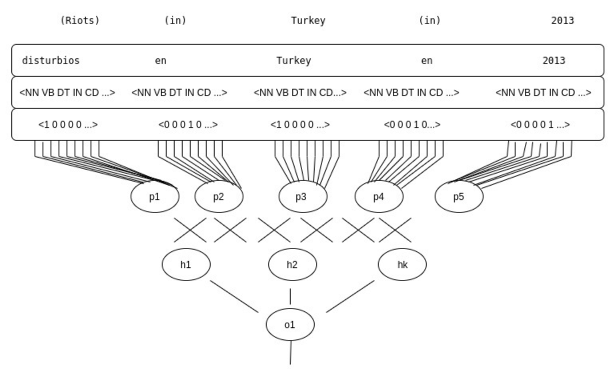
  1. (Subject-action-object) triplet extraction from Dependency Parse Tree. We devised linguistic rules to further narrow down the output event phrase candidates to those which are syntactically correct and semantically relevant to the article. One such example of linguistic rule: First set the verb in the tree as the "head" and traverse the tree to determine:

     – The subject subtree connected to the verb with NSUB or NSUBPASS link

     – The object subtree connected to the verb by DOBJ or POBJ link

     We prioritize these event phrase candidates and relevant terms by assigning higher weight to terms with higher TF-IDF scores or with named entities. (Since phrases with named entities are more likely to be important.) . After this final filter, we feed the list of events to the query system for summary retrieval.

  2. Neural Networks: We trained deep neural networks to identify the linguistic structure of commonly tagged events. The linguistic structure is defined by the combination of POS tags and word relationships found in the tagged event phrases. We began with the simple 1 node neural network (a single perceptron) and tested out neural networks with varying complexities and gave as input a binary vector representing the POS structure of the training 5-gram phrase input.



◦

Figure 1: Neural Net

Each token in a candidate n-gram is represented as a vector that describes linguistic structure by the following schemes:

  - Flipping the bit corresponding to the POS tag for the current word word
  - OR Encoding the word-distance in a sentence of the current word to each POS tag

  These vectors are fed as input and using the backpropagation training algorithm, we allow the neural network to train its edge weights. Once trained, we test it by taking every n-gram from an article and feeding it into this neural network which outputs the top k likely events from an ordering computed over the neural network score. We experiment with different parameters and varying sizes of layers of neural networks to fine tune our framework for best possible results. We do this by measuring the precision, recall and f1 score for the list of events discriminated.

3. Combination of 1 and 2 (Where we used both the dependency parser and the neural network for further filtering and event selection).

- We then use these event-keyphrases extracted above to query search engines and compile a summary from snippets extracted across different news sources. For this we crawl the web and retrieve those articles that hold high relevance score with the query terms. Once this data is gathered, we rank sentences in each of these retrieved articles and return summary comprising of top k-ranked sentences from each source (where k can be adjusted according to the user's requirement for reading length). This ranking uses cosine similarity between the query-tf-idf-vector and snippet-sentence-vector as a metric to gauge relevance to the query and sentence being ranked. These vectors are generated using the TFIDF-vectorizer.

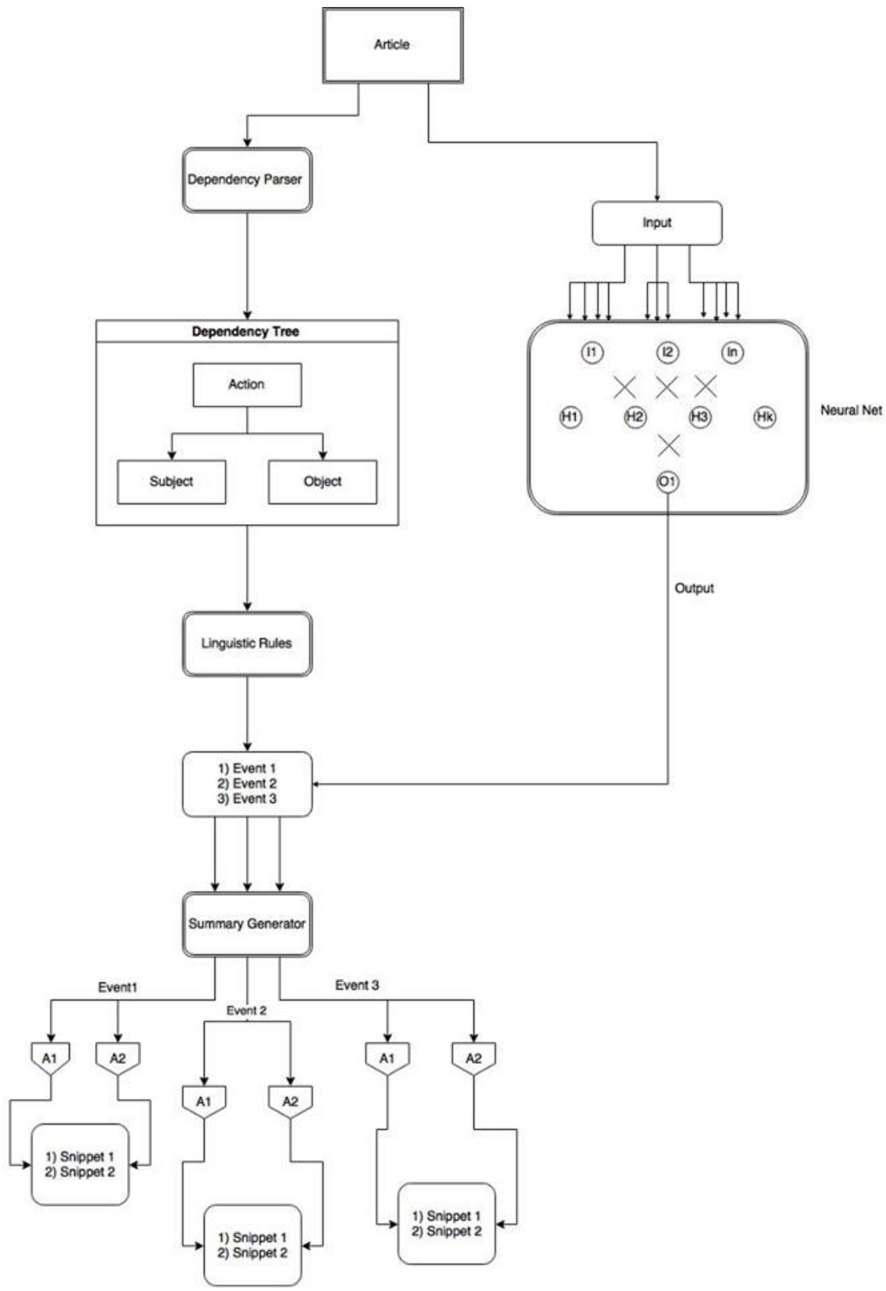


Figure 2: Flow Diagram

4 Evaluation

All team members annotated Spanish news articles by tagging only those phrases as events which adhere to the event definition mentioned in the introduction section. The agreement between team members came out to be 87%. Since the system accuracy cannot be better than human judgment, we use this as our benchmark.

The effectiveness of dependency parser and neural network has been evaluated using the F1 measure. The F1-measure is the standard metric adopted by NLP community for event detection tasks. In our case, the F1-measure of an event detection system is the accuracy of the system to detect events from Spanish news articles. We define the individual sub-measures as follows:

- True positives (tp) - the number of events present in both, the hand tagged test article and the list of events generated events.
- False positives (fp) - the number of events which were present in the system generated events but not in the hand tagged article.
- False negatives (fn) - the number of events which were present in the hand tagged article but not in the system generated events.
- True negatives (tn) - the number events which were not detected by system and were not tagged in the article as well.

$$precision = \frac{tp}{tp + fp} \qquad recall = \frac{tp}{tp + fn}$$

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

The baseline for neural network approach is a single perceptron and for dependency parser is a simple verb phrase extractor. We compare our system's performance to the agreement achieved while annotating test data and with the most sophisticated event-detection systems (like TIPSem)(Kalita, 2016) which like other peer systems returns most generic verb phrases as possible events.

5 Results

Our system achieves the following objectives:

- Detecting and extracting events with high accuracy.
  - We got extremely high recall results by simply using the dependency parser and improved on precision by further implementing linguistic rules and tf-idf. The end result was a framework that gives highly-relevant event-phrases given an input Spanish news article.
  - Categorizing all 5-grams from the input article as events/non-events using neural networks . We demonstrate that a neural network can be used as a classifier for linguistic phrases such as events. The precision is not as good as that of the dependency parser which can be attributed to :
    - the dearth of training data.
    - input model: 5-grams representation for neural network input (5 was the ideal length to capture events). We convert 5-grams into a numeric input vector by flipping bits for each word that correspond to a certain POS tag, causing loss of semantic information. A better numeric/vector representation of candidate phrases for the neural network might yield much better results.(Ceesay & Hou, n.d.)
  - TIPSem is a system which performs a similar task of extracting events from text, though the definition of an event for TIPSem is quite generic. For Spanish, the precision, recall, F1 numbers were for event recognition :0.90, 0.86, 0.88. We have restricted the events to be temporal in nature by picking them from news articles, making the problem tougher. The strategies used by us are not as sophisticated as TIPSem, but still achieved convincing results (to extract only those events which denote a historical occurrence and are relevant to the news subject matter score).
  - The baseline verb phrase extractor has a recall of 1.0 as it lists out all the actions from text and therefore doesn't ignore any tagged event. The precision and F1 score is very low as the number of events extracted by baseline approach is high. Our approach, being selective, ignores many potiential events, So, the recall value is less than one.
- Event information Summary generator: The framework is able to use the Microsoft Bing API to fetch information from web pages about extracted events. We use our query framework to retrieve the most relevant snippets from the returned web pages and combine these snippets for a condensed summary. Most general summarizers in the field of NLP shorten and summarize a given text. However, we generate a summary of the background

information on the events which might not be present or explained in the given article itself.

Table 1: Precision, recall and F1 score for four strategies

	Test data		
	precision	recall	F1
Dependency parser - (subject action object)	0.5617	0.6938	0.6208
Dependency parser - (subject action object) + (action object)	0.7138	0.8922	**0.793**
Neural network - top 3 candidates	0.6577	0.5712	0.6114
Neural network - top 7 candidates	0.6373	0.7136	0.6732
Baseline - verb phrase	0.2857	1.0	0.444

6 Discussion

Experimenting with neural networks and dependency parsing, we discovered concrete strategies for event detection. Some conclusions we found:

- Training the language model for a neural network with domain specific data and testing cases from that domain significantly improves performance (f1 score). Increasing the number of nodes in the hidden layer also gives better accuracy.

- A rule based event generator that uses entities from a dependency parser significantly outperforms neural network trained over POS tags to build a language model for event-phrases when it comes to recognizing events.

The implications that we've deduced include:

- Domain specific training of neural networks gives much better performance over general purpose training when the language model models the POS structure of event phrases

- The amount of training data available has significant impact on the accuracy.

- Naively returning concatenation of (Subject-tree, Verb, Object-tree) triplets extracted from dependency parser gives great recall but low precision.Improving precision requires being selective from these candidates which are then filtered by TF-IDF scores and importance rules such as prioritizing those phrases which have named entities.

We provide a practical application which allows users to conveniently retrieve further information about events from an article. We contribute to the NLP community by providing a library that can be used for event-detection in Spanish and a method to compare and contrast text from different sources specific to a given phrase. For example, one can compare credibility and sentiment across various news sources that are reporting on a single event.

Further research can be focused on :

1. Testing different language models for the neural network. Instead of a binary representation of POS-tag structure, the input vectors can represent the text data through other methods (such as describing the linguistic relationships in between the training words).

2. Different language specific linguistic rules for selecting the best candidates as "event phrases" and using these rules to create event phrases from subject-verb-object entities extracted from the dependency parser.

References

Ceesay, B., & Hou, W.-J. (n.d.). Event extraction and classification by neural network model.

Google. (n.d.). Cloud natural language api. Retrieved from https://cloud.google.com/natural-language/

Kalita, J. (2016). Detecting and extracting events from text documents. *CoRR*, *abs/1601.04012*. Retrieved from http://arxiv.org/abs/1601.04012

Stanford. (n.d.). stanford-postagger-full-2016-10-31. Retrieved from https://nlp.stanford.edu/software/tagger.shtml

TempEval-2. (n.d.). Evaluating events, time expressions, and temporal relations. Retrieved from http://www.timeml.org/tempeval2/

University, B. Y. (n.d.). Spanish corpus for calculation of tfidf. Retrieved from corpusdata.org

Yang, B., & Mitchell, T. M. (2016). Joint extraction of events and entities within a document context. *CoRR*, *abs/1609.03632*. Retrieved from http://arxiv.org/abs/1609.03632