**Event Graph Based multi-document Summarization**

Aastha Sanghi (2016H1120159P)

**Problem Statement**

The aim of this project is to summarize a collection of related documents and provide query based relevant information. Given a group of related documents (e.g., news stories) and the word limit of the summary, the system’s task is to recognize the most relevant information from these documents and avoid any redundancy in the resulting summary. An extractive multi-document summarization model is used which selects sentences based on the relevance of the individual event mentioned in that sentence and also the temporal structure of events. The performance of the implemented summarization method is calculated using the standard ROUGE metric suite.

**Background of the problem**

1. **Description of the selected application domain**

DUC-2004 news articles which consist of several related news articles on a topic. Our system analyzes the multi document on given query and summarizes it and returns a summary file.

1. **Motivation of the problem**

The primary objective is that of designing, implementing and evaluating news domain, query-based Multi-document summarization system which is capable of compiling an acceptably-coherent information from the most related sources of news, whilst making it easy for the reader to access the relevant information as per the query.

1. **Technical issues included in your work.**

ClearTK framework consists of models which are trained on huge news corpus(timebank) and these models are used in our project to further extract events and temporal relations among them. NLTK for tokenization, POSTAG, stemming and lemmatization. Dom parser for parsing xml documents.

**Related Work**

Taking a broader view, many IR models have been proposed that use structured document representations that are not based on events, e.g. (Pai, Chen, Chu, & Chen, 2013; Sbattella & Tedesco, 2013). Pai et al. (2013) structure documents as sets of subject-predicate-object triples, the so-called *content maps*. The similarity between two documents is determined by comparing their content maps, while the triples themselves are compared by comparing the words from the corresponding syntactic categories (e.g., a subject from one triple is compared to a subject in another triple). Although in some aspects similar to our work, triples that Pai et al. consider in many cases do not correspond to event mentions. Furthermore, their model only accounts for the similarity between pairs of documents and not between a document and a query. As regards the performance, no firm conclusions can be drawn because of the absence of comparison with traditional IR paradigms. Sbattella and Tedesco (2013) represent documents at two levels: the conceptual and lexical level. At the conceptual level, they tag the documents with the concepts from a domain ontology, whereas at the lexical level they associate the words with synonym sets from WordNet. Their model is capable of handling keyword-based queries but also more complex natural language queries. However, at the conceptual level their model relies on a domain ontology, which is often not available in ad hoc information retrieval settings. Unlike the event-based IR model we present, the model of Sbattella and Tedesco (2013) employs a shallow-structured document representation that does not incorporate any event-specific semantics.

**Methodology**

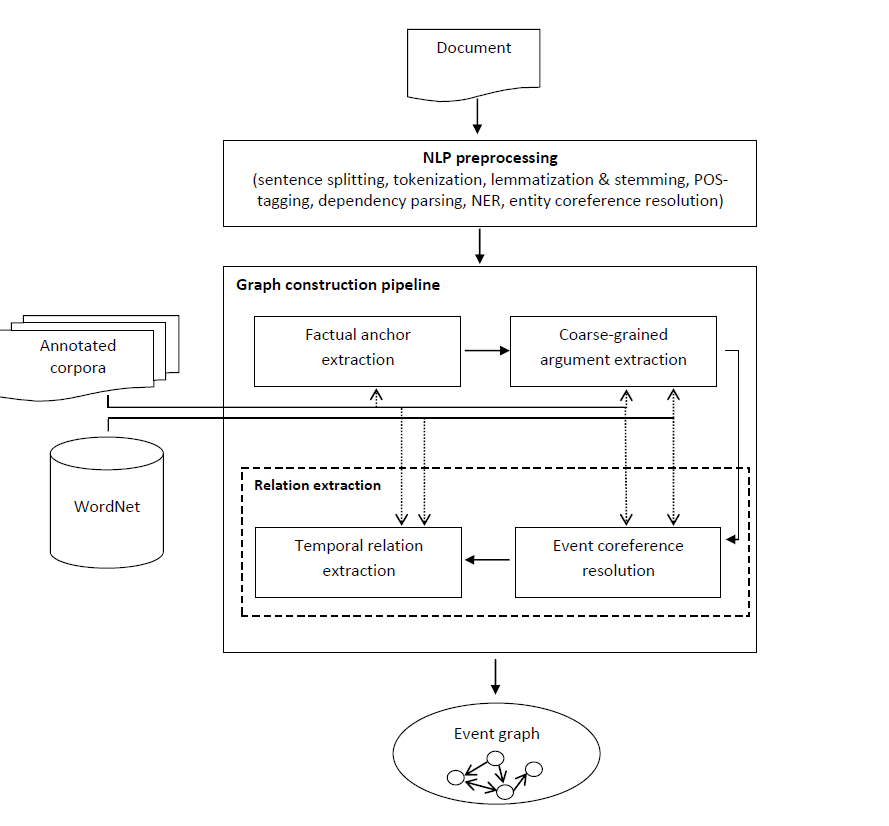
**Construction of event graph:**

**Module 1: Extraction of events and arguments**

1. We address the task of event anchor extraction in two steps: anchor identification and event type classification. For each token, anchor identification predicts whether it is an event.
2. The baseline model predicts for each word the most frequent class
3. assigned to that word on the training set. If the word was never observed in the training set, it is assigned the most frequent class in the training set: class Not event and class Occurrence for the anchor identification and event type classification task, respectively.
4. Argument extraction consists of two steps: (1) argument detection, which syntactically identifies argument candidates, and (2) argument type classification.

**Module 2: Finding temporal relations among events**

1. Positional features measure the distance between and relative position of two event anchors.
2. They include a feature indicating whether the event anchors are in the same sentence, a feature indicating whether the anchors are in adjacent sentences.
3. Lexical features include the word, lemma, stem, and POS of both event Anchors.
4. Syntactic features are computed only for event pairs from the same sentence and include the syntactic path between the anchors (concatenation of the dependency relation labels on the path between the anchors in the dependency tree) and binary features indicating whether the first event anchor syntactically dominates the second and vice versa.



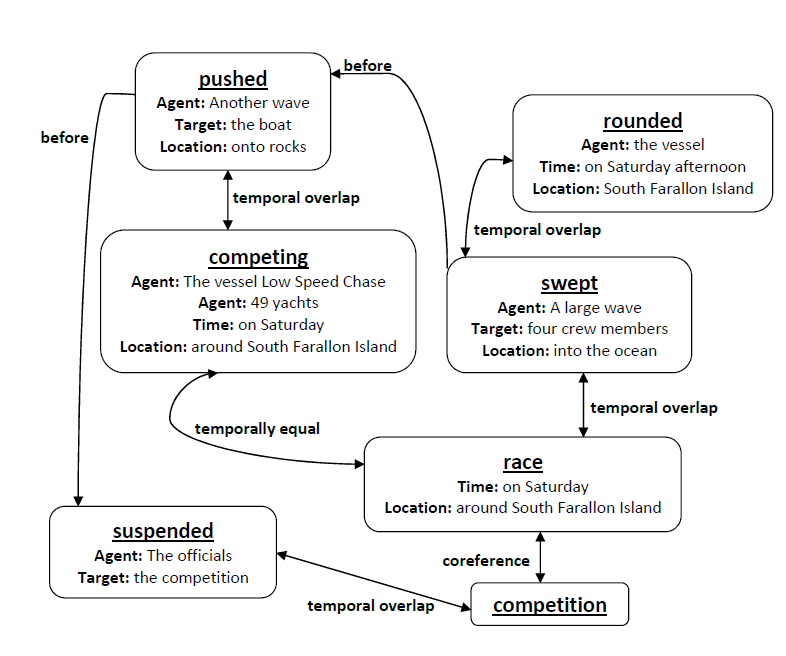
**Summarization:**

**Module 3: Construction of Product kernel**

1. We use the conormal product of graphs as a mechanism to capture exact overlaps between the constructed two event graphs.
2. A graph kernel is akernel function that computes an inner product on graphs. Graph kernels can be intuitively understood as functions measuring the similarity of pairs of graphs.
3. Similarity is conducted based on events and their class. And also the arguments (agent and target) related to it.
4. The labels of vertices v ∈ V (GC ) and v ∈ V (GR) match, denoted L(v) =L(v), iff the anchors and all arguments of event mentions m(v) and m(v) match. The anchor of m(v) and anchor of m(v) match iff they point to the same token from text and have the same event type (e.g., Occurrence). Construction and Evaluation of Event Graphs of arguments of m(v) and m(v) match iff they are identical, i.e., have the same head word and same type;
5. The labels of edges (v1, v2) ∈ E(GC ) ∪ A(GC ) and (v1, v2) ∈ E(GR) ∪
6. A(GR) match iff the corresponding relations are identical, i.e., L(v1, v2) =L(v1, v2) ⇔ r(v1, v2) = r(v1, v2).

**Module 4: Summarization and evaluation**

1. The initial step is the construction of event graphs for all documents from a group of topically related documents.
2. Then compute the relevance score of each pair of event mention from the event graph based on three criteria: (1) the similarity of the event’s participants, (2) the informativeness of the event, and (3) the temporal relations among the events.
3. Then use the temporal structure of the event graphs to refine both scores using the PageRank algorithm.
4. We rank the event mentions according to the refined scores and compute the final score for each event mention as the sum of its ranks in both rankings.
5. After obtaining the final scores for the individual event mentions, we compute the scores for each sentence by summing the scores of the event mentions it contains.
6. To avoid redundancy, we additionally cluster the selected sentences based on semantic similarity.
7. Finally, we select the representative sentences from each semantic cluster to constitute the summary.



**Experimental Analysis**

**Event class:**

|  |  |  |
| --- | --- | --- |
| **Event Class** | **Dataset1** | **Dataset2** |
| Occurrence | 412 | 527 |
| Reporting | 106 | 115 |
| I\_State | 19 | 12 |
| I\_Action | 26 | 23 |
| Aspectual | 7 | 5 |

**Argument extraction:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dataset** | **Agent** | **Target** | **Time** | **Location** |
| **Train** | 1072 | 867 | 212 | 196 |
| **Test** | 356 | 264 | 77 | 65 |

**Event Identification Accuracy (performed on EvExtra corpus):**

|  |  |  |  |
| --- | --- | --- | --- |
| **Anchor Identification** | **Precision** | **Recall** | **F1** |
| **EvGraph-Anchors time bank** | **89** | **64** | **77** |

**Summarization evaluation results:**

|  |  |  |
| --- | --- | --- |
| **Dataset** | **ROUGE-1** | **ROUGE-2** |
| **DUC-2004** | 0.397 | 0.67 |

**References**

1. Glavaš, G., & Šnajder, J. (2014). Event graphs for information retrieval and multi-document summarization. *Expert systems with applications*, *41*(15), 6904-6916
2. Pai, M. Y., Chen, M. Y., Chu, H. C., & Chen, Y. M. (2013). Development of a semantic-based content mapping mechanism for information retrieval. *Expert Systems with Applications*, *40*(7), 2447-2461.
3. Sbattella, L., & Tedesco, R. (2013). A novel semantic information retrieval system based on a three-level domain model. *Journal of Systems and Software*, *86*(5), 1426-1452
4. https://cleartk.github.io/cleartk/docs/index.html
5. Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".