Building Event Threads out of Multiple News Articles

Xavier Tannier
LIMSI-CNRS
Univ. Paris-Sud
Orsay, France
xavier.tannier@limsi.fr

Véronique Moriceau
LIMSI-CNRS
Univ. Paris-Sud
Orsay, France
moriceau@limsi.fr

Abstract

We present an approach for building multi-document event threads from a large corpus of newswire articles. An event thread is basically a succession of events belonging to the same story. It helps the reader to contextualize the information contained in a single article, by navigating backward or forward in the thread from this article. A specific effort is also made on the detection of reactions to a particular event.

In order to build these event threads, we use a cascade of classifiers and other modules, taking advantage of the redundancy of information in the newswire corpus.

We also share interesting comments concerning our manual annotation procedure for building a training and testing set[1].

1 Introduction

In this paper, we explore a new way of dealing with temporal relations between events. Our task is somewhat between multidocument summarization and classification of temporal relations between events. We work with a large collection of English newswire articles, where each article relates an event: the main topic of the article is a specific event, and other older events are mentioned in order to put it into perspective. Thus, we consider that an event is associated with an article and that defining temporal relations between articles is a way to define temporal relations between events.

[1]This work has been partially funded by French National Research Agency (ANR) under project Chronolines (ANR-10-CORD-010). We would like to thank the French News Agency (AFP) for providing us with the corpus.

The task is to build a temporal graph of articles, linked between each other by the following relations:

- *Same event*, when two documents relate the same event, or when a document is an update of another one.
- *Continuation*, when an event is the continuation or the consequence of a previous one.

We also define a subset of *continuation*, called *reaction*, concerning a document relating the reaction of someone to another event.

Some examples of these three classes will be given in Section 3.

These relations can be represented by a directed graph where documents are vertices and relations are edges (as illustrated in all figures of this article). Figure 1 shows an example of such a graph.

Press articles, and especially newswire articles, are characterized by an important redundancy of related events. An important event[2] is likely to be treated by several successive articles, which will give more and more details and update some numbers (mainly, tragedy casualty updates, as shown in Figure 2). On the one hand, this redundancy is an issue since a system must not show duplicate information to the user; on the other hand, we show in this article that it can also be of great help in the process of extracting temporal graphs.

In what follows, we first review some of the related work in Section 2. Section 3 presents the annotation procedure and the resulting annotated corpus

[2]Note that we do not focus intentionally on "important" events. However, the fact is that minor events do hardly lead to dense temporal graphs.

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 958–967,
Seattle, Washington, USA, 18-21 October 2013.

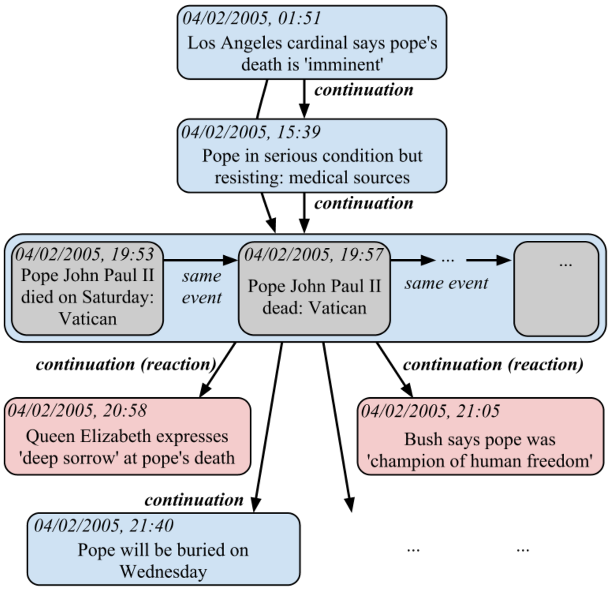

Figure 1: Example of "temporal graph": around the Pope's death. The associated text is the title of each article. Relations that can be obtained by transitivity have been hidden for clarity's sake.

used for developing, learning and evaluating the system. The simple modules used to predict the *same event*, *continuation* and, possibly, *reaction* relations are described in Section 4, and results are given in Section 5.

We also propose an end-user application to this work. When a user reads an article, the system will then be able to provide her with a thread of events having occurred before or after, helping her to contextualize the information she is reading. This application is described in Section 6.

2 Related work

The identification of temporal relations between events in texts has been the focus of increasing attention because of its importance in NLP applications such as information extraction, question-answering or summarization. The evaluation campaigns TempEval 2007 (Verhagen et al., 2007) and TempEval 2010 (Verhagen et al., 2010) focused on temporal relation identification, mainly on temporal relations between events and times in the same sentence or in consecutive sentences and between events and the creation time of documents. In this context, the goal is to identify the type of a temporal relation which is

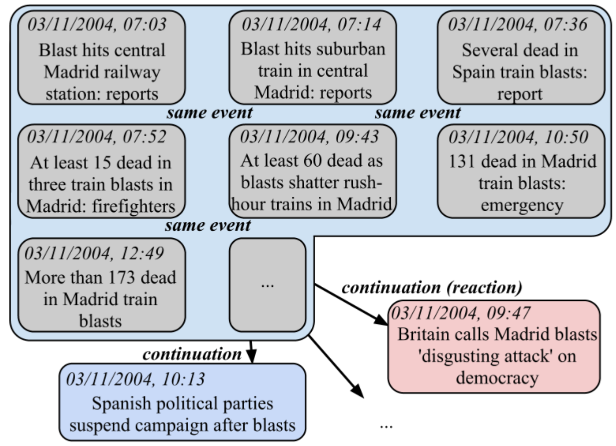

Figure 2: Example of "temporal graph": Madrid attacks, with many updates of the initial information. Note that articles gathered in this main pool of articles can be posterior to the continuations and reactions to the described event.

known to be present. Systems having the best results (accuracy about 0.6) use statistical learning based on temporal features (modality, tense, aspect, etc.) (Mani et al., 2006; Chambers et al., 2007). More recently, Mirroshandel and Ghassem-Sani (2012) proposed a new method for temporal relation extraction by using a bootstrapping method on annotated data and have a better accuracy than state-of-the-art systems. Their method is based on the assumption that similar event pairs in topically related documents are likely to have the same temporal relations. For this work, the authors had already some collections of topically related documents and did not need to identify them.

In the 2012 i2b2 challenge (i2b, 2012), the problem was not only to identify the type of temporal relations, but also to decide whether a temporal relation existed or not between two elements, either clinical concepts or temporal expressions. But, as in TempEval, the temporal analysis were only to be performed within a single document.

Other works focus on event ordering. For example, Fujiki et al. (2003) and Talukdar et al. (2012) proposed methods for automatic acquisition of event sequences from texts. They did not use temporal information present in texts and extracted sequences of events (*e.g.* *arrest/escape*) from sentences which were already arranged in chronological order. Chambers and Jurafsky (2008) proposed a method to learn narrative chains of events related to a protagonist in a single document. The first step consists in detecting narrative relations between events sharing coreferring arguments. Then, a temporal classifier orders partially the connected events with the *before* relation.

Concerning the identification of the *reaction* relation, to our knowledge, there is no work on the detection of reaction between several documents. Pouliquen et al. (2007), Krestel et al. (2008) and Balahur et al. (2009) focused on the identification of reported speech or opinions in quotations in a document, but not on the identification of an event which is the source of a reaction and which can possibly be in another document.

As we can see, all these approaches, as well as traditional information extraction approaches, lean on information contained by a single document, and consider an event as a word or a phrase. However, Ahmed et al. (2011) proposed a framework to group temporally and tocipally related news articles into *same story* clusters in order to reveal the temporal evolution of stories. But in these topically related clusters of documents, no temporal relation is detected between articles or events except chronological order. On this point of view, our task is closer to what is done in multidocument summarization, where a system has to detect redundant excerpts from various texts on the same topic and present results in a relevant chronological order. For example, Barzilay et al. (2002) propose a system for multidocument summarization from newswire articles describing the same event. First, similar text units from different documents are identified using statistical techniques and shallow text analysis and grouped into thematic clusters. Then, in each theme, sentences which are selected as part of the summary are ordered using the publication date of the first occurrence of events to order sentences.

3 Resources

We built an annotated collection of English articles, taken from newswire texts provided by the French news agency (AFP), spreading over the period 2004-2012. The entire collection contains about 1.5 million articles. Each document is an XML file containing a title, a creation time (DCT), a set of keywords and textual content split into paragraphs.

3.1 Selection of Article Pairs

Pairs of documents were automatically selected according to the following constraints:

- The article describes an event. Articles such as timelines, fact files, agendas or summaries were discarded (all these kinds of articles were tagged by specific keywords, making the filtering easy).
- The distance between the two DCTs does not exceed 7 days.
- There are at least 2 words in common in the set of keywords and/or 2 proper nouns in common in the first paragraph of each article.

These last two restrictions are important, but necessary, in order to give annotators a chance to find some related articles. Pure random selection of pairs over a collection of 1.5 million articles would be impractical.

We assume that the title and the first paragraph describe the event associated with the document. This is a realistic hypothesis, since the basic rules of journalism impose that the first sentence should summarize the event by informing on the “5 Ws” (*What*, *Who*, *When*, *Where*, *Why*). However, reading more than the first paragraph is sometimes necessary to determine whether a relation exists between two events.

3.2 Relation Annotation

Two annotators were asked to attribute the following relations between each pair of articles presented by the annotation interface system.

In a first annotation round, 7 types of relations were annotated:

- Three relations concerning cases where the two articles relate the same event or an update:
  - *number update*, when a document is an update of numerical data (see top of Figure 5),
  - *form update*, when the second document brings only minor corrections,

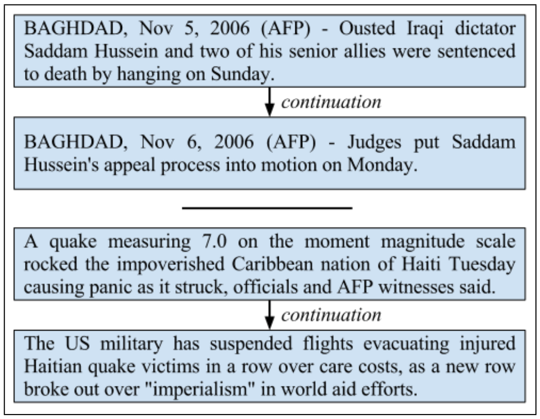

Figure 3: Examples of relation *continuation* between two documents.

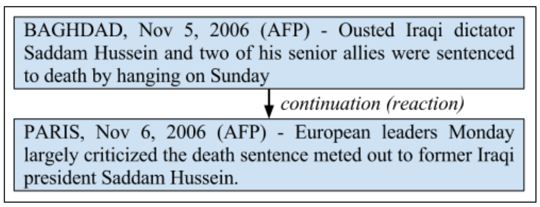

Figure 4: Examples of relation *continuation-reaction* between two documents.

  – *details*, when the second document gives more details about the events (see bottom of Figure 5).

- *development of same story*, when the two documents relate two events which are included into a third one;

- *continuation*, when an event is the continuation or the consequence of a previous one. Figure 3 shows two examples of such a relation. It is important to make clear that a *continuation* relation is more than a simple thematic relation, it implies a natural prolongation between two events. For example, two sport events of the same Olympic Games, or two different attacks in Iraq, shall not be linked together unless a direct link between both is specified in the articles.

- *reaction*, a subset of *continuation*, when a document relates the reaction of someone to another event, as illustrated by the example in Figure 4.

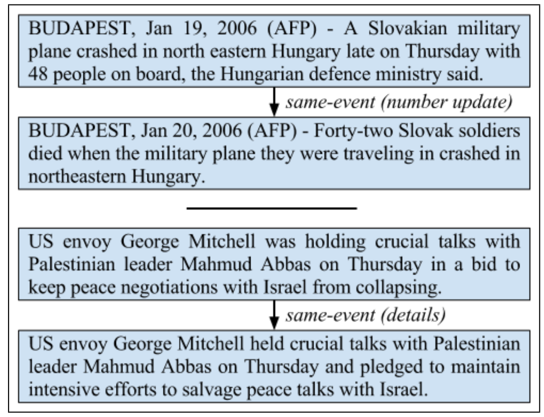

Figure 5: Example of relations *same-event* between two documents: update on casualties (top) or details (bottom).

- *nil*, when no relation can be identified between the two documents.

The inter-annotator agreement was calculated with Cohen's Kappa measure (Cohen, 1960) across 150 pairs: $\kappa = 0.68$. The agreement was low for the first 4 types of relations mostly because the difference between relations was not clear enough. We therefore aggregated the *number update*, *form update* and *details* relations into a more generic and consensual *same-event* relation (see Figure 5). We also discarded the *development of same story* relation, leaving only *same-event*, *continuation* and *reaction*.

Annotation guidelines were modified and a second annotation round was carried out: only the *same-event*, *continuation*, *reaction* and *nil* relations were annotated. Inter-annotator agreement across 150 pairs was then $\kappa = 0.83$, which is a good agreement.

3.3 Relation Set Extension

This manual annotation would have led to very sparse temporal graphs without the two following additional processes:

- When the annotator attributed a "non-nil" relation to a pair of documents, the annotation system suggested other pairs to annotate around the concerned articles.

- *Same-event* and *continuation* relations are transitive: if A *same-event* B and B *same-event* C, then A *same-event* C (and respectively for

	Pair number	Learning	Evaluation
Same event	762	458	304
Continuation	1134	748	386
Reaction	182	123	59
Nil	918	614	304
TOTAL	**2996**	1943	1053

Table 1: Characteristics of the corpus.

continuation). Then, when the annotation was done, a transitive closure was performed on the entire graph, in order to get more relations with low effort (and to detect and correct some inconsistencies in the annotations).

Finally, almost 3,000 relations were annotated. 2⁄3 of the annotated pairs were used for development and learning phases, while 1⁄3 were kept for evaluation purpose (cf. Table 1).

4 Building Temporal Graphs

As we explained in the introduction, the main purpose of this paper is to show that it is possible to extract temporal graphs of events from multiple documents in a news corpus. This is achieved with the help of redundancy of information in this corpus. Therefore, we will use a cascade of classifiers and other modules, each of them using the relations deduced by the previous one. All modules predict a relation between two documents (*i.e.*, two events).

We did not focus on complex algorithms or classifiers for tuning our results, and most of our features are very simple. The idea here is to show that good results can be obtained in this original and useful task. The process can be separated into 3 main stages, illustrated in Figure 6:

A. Filtering out pairs that have no relation at all, *i.e.* classifying between *nil* and *non-nil* relations;

B. Classifying between *same-event* and *continuation* relations;

C. Extracting *reactions* from the set of *continuation* relations.

All described classifiers use SMO (Platt, 1998), the SVM optimization implemented into Weka (Hall et al., 2009), with logistic models fitting (option "-M"). With this option, the confidence score of each

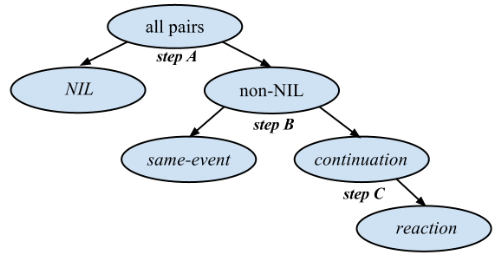

Figure 6: A 3-step classification.

prediction can be used, while SMO alone provides a constant probability for all instances.

From now on, when considering a pair of documents, we will refer to the older document as document 1, and to the more recent one as document 2. The relations found between documents will be represented by a directed graph where documents are vertices and relations are edges.

4.1 A. Nils *versus* non-nils

We first aim at separating *nil* relations (no relation between two events) from other relations. This step is achieved by two successive classifiers: the first one (A.1) uses mainly similarity measures between documents, while the second one (A.2) uses the relations obtained by the first one.

4.1.1 Step A.1: Nil classifier, level 1

Features provided to the SMO classifier at this first step are based on 3 different similarity measures applied to pairs of titles, pairs of first sentences, and pairs of entire documents: cosine similarity (as implemented by Lucene search engine[3]), inclusion similarity (rate of words from element 1 present in element 2) and overlap similarity (number of words present in both elements). This classifier is therefore based on only 9 features.

4.1.2 Step A.2: Nil classifier, level 2

Finding relations on a document implies that the described event is important enough to be addressed by several articles (*same-event*) or to have consequences (*continuation*). Consequently, if we find such relations concerning a document, we are more likely to find more of them, because this means that

[3] `http://lucene.apache.org`

the document has some importance. A typical example is shown in Figure 7, where an event described by several documents (on the left) has many continuations. For this reason, we build a second classifier A.2 using additional features related to the relations found at step A.1:

- Number of non-nil edges, incoming to or outgoing from document 1 (2 features); the sum of both numbers (1 extra feature);

- Number of non-nil edges, incoming to or outgoing from document 2 (2 features); the sum of both numbers (1 extra feature);

- Number of non-nil edges found involving one of the two documents (*i.e.*, the sum of all edges described above – 1 feature).

These figures have been computed on training set for training, and on result of step A.1 classifier for testing. This new information will basically help the classifier to be more optimistic toward non-nil relations for documents having already non-nil relations.

4.2 B. Same-event *versus* Continuation

We are now working only with *non-nil* relations (even if some relations may switch between *nil* and *non-nil* during the transitive closure).

4.2.1 Step B.1: Relation classifier, level 1

Distinction between *same-event* and *continuation* is made by the following sets of features:

- Date features:
  - Difference between the two document creation times (DCTs): difference in days, in hours, in minutes (3 features);
  - Whether the creation time of doc. 1 is mentioned in doc. 2. For this purpose, we use the date normalization system described in Kessler et al. (2012).
  - Cosine similarity between the first sentence of doc. 1 and sentences of doc. 2 containing the DCT of doc. 1.
  - Cosine similarity between the first sentence of doc. 1 and the most similar sentence of doc. 2.

    These last three features come from the idea that a *continuation* relation can be made explicit in text by mentioning the first event in the second document.

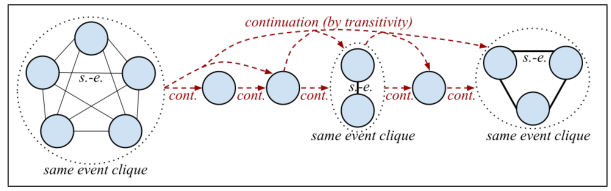

Figure 7: An example of highly-connected subgraph, corresponding to the development of an important story. Same events are grouped by cliques (see Section 4.2.3) and some redundant relations are not shown for clarity's sake.

- Temporal features: whether words introducing temporal relations occur in document 1 or document 2. These manually-collected words can be prepositions (*after, before*, etc.) or verbs (*follow, confirm, prove*, etc.).

- Reaction features: whether verbs introducing reactions occur in document 1 or document 2 (25 manually-collected verbs as *approve, accept, welcome, vow*, etc.).

- Opinion features: whether opinion words occur in document 1 or document 2. The list of opinion words comes from the MPQA subjectivity lexicon (Wilson et al., 2005).

Only *same-event* relations classified with more than 90% confidence by the classifier are kept, in order to ensure a high precision (recall will be improved at next step). This threshold has been set up on development set.

4.2.2 Step B.2: Relation classifier, level 2

As for step A.2, a second classifier is implemented, using the results of step B.1 with the same manner as A.2 uses A.1 (collecting numbers of *same-event* and *continuation* relations that have been found by the previous classifier).

4.2.3 Steps B.3 and B.4: Transitive closure by vote

As already stated, *same-event* and *continuation* relations are transitive. *Same-event* is also symmetric (A *same-event* B $\Rightarrow$ B *same-event* A). In the graph formed by documents (vertices) and relations (edges), it is then possible to find all cliques, *i.e.* subsets of vertices such that every two vertices in the subset are connected by a *same-event* relation, as illustrated by Figure 7.

This step does not involve any learning phase. Starting from the result of last step, we find all *same-event* cliques in the graph by using the Bron and Kerbosch (1973) algorithm. The transitive closure process is then illustrated by Figure 8. If the classifier proposed a relation between some documents of a clique and some other documents (as D1, D2 and D3), then a vote is necessary:

- If the document is linked to half or more of the clique, then all missing links are created (Figure 8.a);
- Otherwise, the document is entirely disconnected from the clique (Figure 8.b).

This vote is done for *same-event* and *continuation* relations (resp. steps B.3 and B.4). Only cliques containing at least 3 nodes are used. A drawback of this voting procedure is that the final result may not be independent of the voting order, in some cases. However, it is assured that the result is consistent, *i.e.* that no document will sit in two different cliques, or that two documents from the same clique will not have two different relations toward a third document.

Note that this vote leads to improvements only if the precision of the initial classifier is sufficiently good. As we will see in Section 5.2, this is the case in our situation, but one must keep in mind that a vote leaning on too imprecise material would lead to even worse results. Some experiments on the development set show us that at least 70% precision was necessary. Another way to ensure robustness of the vote would be to apply the transitive closure only on bigger cliques (*e.g.*, containing more than 3 or 4 nodes).

4.3 C. Continuation *versus* Reaction

The approach for reaction extraction is different. We first try to determine which documents describe reactions, regardless of which event it is a reaction to. In the training set, all documents having at least one incoming *reaction* edge are considered as reaction

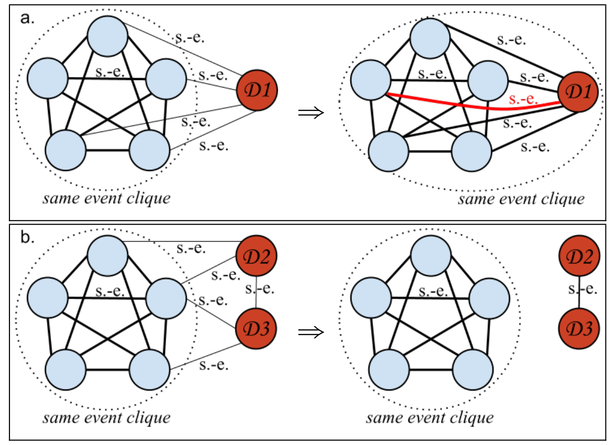

Figure 8: Vote for *same-event* transitive closure. At the top (a.), four nodes from the 5-node clique are linked to document D_1, which is enough to add D_1 to the clique. At the bottom (b.), only two nodes from the clique are linked to documents D_2 and D_3, which is not enough to add them into the clique. All edges from the clique to D_2 and D_3 are then deleted.

documents, all others are not. This distinction is then learned with the same model and features as for step B.1 (Section 4.2.1).

Once reaction documents have been selected, the question is how to decide to which other document(s) it must be linked. For example, in Figure 1, *"Queen Elizabeth expresses deep sorrow"* is a reaction to pope's death, not to other documents in the temporal thread (for example, not to other reactions or to *"Pope in serious condition"*). We did not manage to build any classifier leading to satisfying results at this point. We then proposed the two following basic heuristics, applied on all *continuation* relations found after step B:

- A reaction reacts to only one event.
- A reaction reacts to an important event. Then, among all *continuation* edges incoming to the reaction document, we choose the biggest *same-event* clique and create *reaction* edges instead of *continuations*. If there is no clique (only single nodes) or several same-size cliques, all of them are tagged as *reactions*.

This module is called step C.1. Finally, a transitive closure is performed for reactions (C.2).

Relation	Precision	Recall	F1
NIL	0.754	0.821	0.786
same-event	0.832	0.812	0.822
continuation	0.736	0.696	0.715
↳ *reaction*	0.273	0.077	0.120

Table 2: Results obtained by the baseline system. *Continuation* scores do not consider reactions, only the last row makes the distinction.

5 Results

5.1 Baseline

As a baseline, we propose a single classifier determining all classes at once, based on the same SMO classifier with the exact same parameters and all similarity-based features (on titles, first sentences and entire documents) described in Section 4.1.1.

Table 2 shows results for this baseline. Unsurprisingly, *same-event* relations are quite well classified by this baseline, since similarity is the major clue for this class. *Continuation* is much lower and only 3 reactions are well detected.

5.2 System Evaluation

Results for all successive steps described in previous section are shown in Figure 3. The final result of the entire system is the last one. The first observation is that redundancy-based steps improve performance in a significant manner:

- Classifiers A.2 and B.2, using the number of incoming or outgoing edges found at previous steps, lead to very significant improvement.
- Among transitivity closure algorithms (B.3, B.4, C.2), only *same-event* transitivity B.3 leads to significant improvement. Furthermore, as we already noticed, these algorithms must be used only when a good precision is guaranteed at previous step. Otherwise, there is a risk of inferring mostly bad relations. This is why we biased classifier at step B.1 towards precision. Finally, if this condition on precision is true, transitivity closure is a robust way to get new relations for free.

Results also tell that classification of relations *same-event* and *continuation* is encouraging. *Reaction* level gets a fair precision but a bad recall. This is not catastrophic since most of the missed reactions are tagged as *continuation*, which is still true (only 10% of the *reaction* relations are mistagged as *same-event*). However, there is big room for improvement on this point.

Step	**Relation**	**Precision**	**Recall**	**F1**
A. *NIL* vs *non-NIL classifier*				
A.1	NIL	0.764	0.815	0.788
	non-NIL	0.921	0.896	0.910
A.2 ***	NIL	0.907	0.811	0.857
	non-NIL	0.925	0.966	0.945
B. *Same-event* vs *continuation classifier*				
B.1	NIL	0.907	0.811	0.857
	same-event	0.870	0.553	0.676
	continuation	0.664	0.867	0.752
B.2 ***	NIL	0.947	0.831	0.885
	same-event	0.894	0.724	0.800
	continuation	0.744	0.911	0.819
B.3 **	NIL	0.884	0.831	0.857
	same-event	0.943	0.819	0.877
	continuation	0.797	0.906	0.848
B.4 *	NIL	0.890	0.831	0.860
	same-event	0.943	0.819	0.877
	continuation	0.798	0.911	0.851
C. *Reaction* vs *continuation*				
C.1 C.2	NIL	**0.890**	**0.831**	**0.860**
	same-event	**0.943**	**0.819**	**0.877**
	continuation	**0.798**	**0.911**	**0.851**
	↳ reaction	**0.778**	**0.359**	**0.491**

Table 3: Results obtained at each step of the classification process. The significance of the improvement wrt previous step (when relevant) is indicated by the Student t-test (*: non significant; **: $p < 0.05$ (significant); ***: $p < 0.01$ (highly significant)). Steps C.1 and C.2 are aggregated, since their results are exactly the same.

6 Application

As we showed in previous section, results for classification of *same-event* and *continuation* relations between documents are good enough to use this system in an application that builds "event threads" around an input document. The use case is the following:

- The reader reads an article (let's say, about the death of John Paul II, article published on Feb. 4th, 2005 (UT) – see Figure 1).
- A link in the page suggests the user to visualize the event thread around this article.

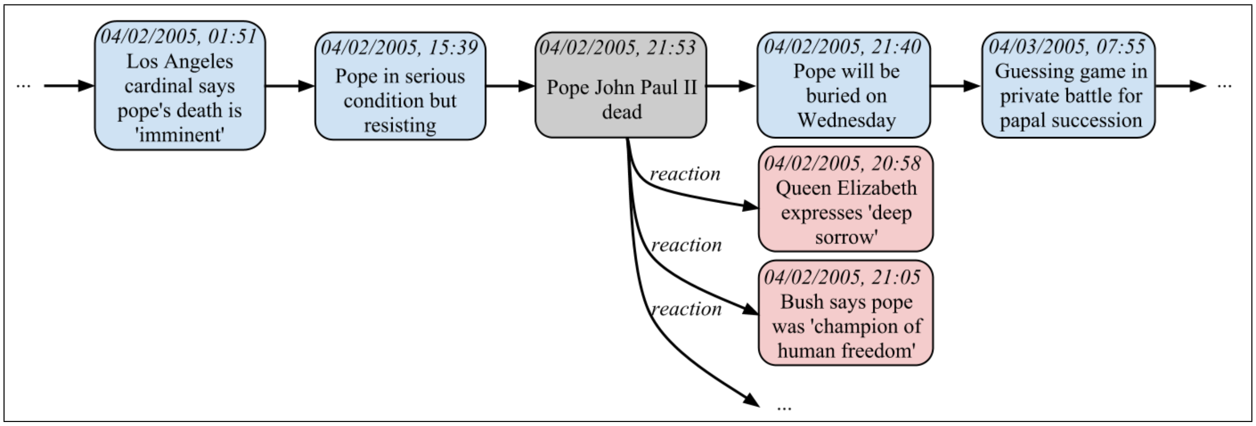

Figure 9: An example of temporal thread obtained on the death of John Paul II for user visualization (see corresponding relation graph in Figure 1).

- All articles within a period of 7 days around the event, sharing at least two keywords with the current document, are collected. All pairs are given to the system[4].

- When *same-event* cliques are found, only the longest article (often, the most recent one) of each clique is presented to the user. However, the date and time presented to the user are those of the first article relating the event.

- This leads to a graph with only *continuation* and *reaction* relations. Edges are "cleaned" so that a unique thread is visible: relations that can be obtained by transitivity are removed, edges between two documents are kept only if no document can be inserted in-between.

- Nodes are presented in chronological order. The user can visualize and navigate through this graph (the event thread shows only titles but full articles can be accessed by clicking on the node).

- When found, reactions are isolated from the main thread.

- Such a temporal thread is potentially infinite. If the user navigates through the end of the 7-day window, the system must be run again on the next time span.

[4]In case of very important events where "all pairs" would be too much, the temporal window is restrained. However, there is no real time performance issue in this system.

Figure 9 presents the result of this process on the partial temporal graph shown in Figure 1.

7 Conclusion

This article presents a task of multidocument temporal graph building. We make the assumption that each news article (after filtering) relates an event, and we present a system extracting relations between articles. This system uses simple features and algorithms but takes advantage of the important redundancy of information in a news corpus, by incorporating redundancy information in a cascade of classifiers, and by using transitivity of relations to infer new links.

Finally, we present an application presenting "event threads" to the user, in order to contextualize the information and recomposing the story of an event.

Now that the task is well defined and that encouraging results have been obtained, we envisage to enrich classifiers by more fine-grained temporal and lexical information, such as narrative chains (Chambers and Jurafsky, 2008) for *continuation* relation or event clustering (Barzilay et al., 2002) for *same-event* relation. There is no doubt that *reaction* detection can be improved a lot, by going beyond simple lexical features and discovering specific patterns. We also intend to adapt the described system to other languages than English.

References

A. Ahmed, Q. Ho, J. Eisenstein, E.P. Xing, A.J. Smola, and C.H. Teo. 2011. Unified Analysis of Streaming News. In *Proceedings of WWW*, Hyderabad, India.

A. Balahur, R. Steinberger, E. van der Goot, B. Pouliquen, and M. Kabadjov. 2009. Opinion Mining on Newspaper Quotations. In *Proceedings of International Joint Conference on Web Intelligence and Intelligent Agent Technologies*, Milano, Italy.

R. Barzilay, N. Elhadad, and K.R. McKeown. 2002. Inferring Strategies for Sentence Ordering in Multidocument News Summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

C. Bron and J. Kerbosch. 1973. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577.

N. Chambers and D. Jurafsky. 2008. Unsupervised Learning of Narrative Event Chains. In *Proceedings of the 46th Annual Meeting of the ACL*, Columbus, USA.

N. Chambers, S. Wang, and D. Jurafsky. 2007. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, Prague, Czech Republic, June.

J. Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 43(6):551–558.

T. Fujiki, H. Nanba, and M. Okumura. 2003. Automatic Acquisition of Script Knowledge from a Text Collection. In *Proceedings of EACL*, Budapest, hungary.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. 2009. The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, 11(1).

2012. *Proceedings of i2b2/VA Shared-Tasks and Workshop on Challenges in Natural Language Processing for Clinical Data*, Chicago, USA.

R. Kessler, X. Tannier, C. Hagège, V. Moriceau, and A. Bittar. 2012. Finding Salient Dates for Building Thematic Timelines. In *Proceedings of the 50th Annual Meeting of the ACL*, Jeju Island, Republic of Korea.

R. Krestel, S. Bergler, and R. Witte. 2008. Minding the Source: Automatic Tagging of Reported Speech in Newspaper Articles. In *Proceedings of LREC*, Marrakech, Morocco.

I. Mani, M. Verhagen, B. Wellner, C. Lee, and J. Pustejovsky. 2006. Machine learning of temporal relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, Sydney, Australia.

S.A. Mirroshandel and G. Ghassem-Sani. 2012. Towards Unsupervised Learning of Temporal Relations between Events. In *Journal of Artificial Intelligence Research*, volume 45.

J.C. Platt, 1998. *Advances in Kernel Methods - Support Vector Learning*, chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization. MIT Press.

B. Pouliquen, R. Steinberger, and C. Best. 2007. Automatic Detection of Quotations in Multilingual News. In *Proceedings of RANLP*, Borovets, Bulgaria.

P.P. Talukdar, D. Wijaya, and T. Mitchell. 2012. Acquiring Temporal Constraints between Relations. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, Hawaii.

M. Verhagen, R. Gaizauskas, F. Schilder, M. Hepple, G. Katz, and J. Pustejovsky. 2007. SemEval-2007 - 15: TempEval Temporal Relation Identification. In *Proceedings of SemEval workshop at ACL*, Prague, Czech Republic.

M. Verhagen, R. Sauri, T. Caselli, and J. Pustejovsky. 2010. SemEval-2010 - 13: TempEval-2. In *Proceedings of SemEval workshop at ACL*, Uppsala, Sweden.

T. Wilson, J. Wiebe, and P. Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of HLT-EMNLP*.