Exploring Social Contexts along the Time Dimension: Temporal Analysis of Named Entities

Brett Walenz, Robin Gandhi, William Mahoney, Quiming Zhu
University of Nebraska at Omaha
Omaha, USA
{bwalenz, rgandhi, wmahoney, qzhu}@unomaha.edu

Abstract— Exploring the evolution of social contexts with time can provide unique insights into human social dynamics. Several social contexts and relationships can be mined from unstructured text articles that describe social phenomena. In contrast to structured graphs of social networks, named entity recognition is a task that attempts to classify elements in unstructured textual items into predefined categories, such as organizations, people, locations, quantities, and temporal expressions. State of the art systems have approached the quality of human annotators on static documents for multiple languages. The problem of constructing and linking corresponding entities across topics and documents still exists. During a temporal sequence, entities fluctuate in frequency over time, and the set of entities in the present document can differ from the beginning and end. Furthermore, with user-generated content available on most major news sites, different viewpoints and entity relationships are generated by different users. This paper describes the Sequencer system for the temporal analysis of named entities in news articles between media reported stories and user generated content.

Natural language processing; social media; temporal extraction

I. INTRODUCTION

Social contexts continuously change with respect to time. Our research effort is geared towards deriving insights from the study of social contexts and their evolution as expressed in unstructured text documents. To derive such insights, the first step to be performed is named entity recognition. It is the process of classifying elements in unstructured textual items into predefined categories, such as organizations, people, locations, quantities, and temporal expressions. Named entity recognition opens up the possibility of extracting rich social contexts from possible countless sources of editorial or informal user generated content, for example web news sites.

State of the art systems have approached the quality of human annotators on static documents for multiple languages. However, the problem of constructing and linking corresponding entities across topics and documents still exists. During a temporal sequence, entities fluctuate in frequency, and the set of entities in the present data can differ from the beginning and end. Furthermore, with user-generated content available on most major news sites, different viewpoints and entity relationships are generated by different users. To extract and study such expression of social contexts, this paper describes the Sequencer system for the temporal analysis of named entities in news articles between media reported stories and user generated content.

News articles generally refer to a temporal event containing the “Five Ws” (who, what, when, where, why) of the event. In today’s media, news articles are also typically sequential in nature, describing the current state of an underlying topic or event. The Sequencer system is a pipelined information extraction system for processing news articles as they occur. It includes the capability of building increasingly complex models of events in news articles over time. Our aim is to model and mine structured information from multiple news sources, as well as differentiate between user-generated content, such as individual news reports from users, to official media reported stories.

Mining, modeling and visualizing the sequential nature of a topic or event theme is an active area of research. Early examples of information visualization systems include Envision[1] and DIVA[2], which are grid or spatial-based representations of documents. ThemeRiver[3] represents thematic changes in a document set by displaying theme terms in a stacked-time-series format. TimeMines[4] constructs timelines from models of word usage, using the named entities as features and assuming each document in the set is explicitly tagged with the date.

However, the systems resulting from these efforts are fairly rigid and unable to work on current news sites, aggregate from multiple sources or respond to frequent updates. Hence, a scalable and highly configurable system that can handle multiple sources as input and respond to quickly changing news topics could be useful for data exploration tasks. Sequencer is our attempt at building such a system to explore the temporal evolution of social contexts in news articles.

Sequencer provides users with temporal views of a sequence of news articles from websites. It is designed to assist users in the identification of patterns, key events, and unexpected shifts in entities and their social contexts within a story. In addition to these key events, most news organizations now offer user-generated viewpoints, such as CNN’s iReport mechanism, where users can submit individual reports often consisting of text, video, and images. In this case, the system aims to offer how the combined social viewpoint of all users relates to the media viewpoint offered by the different news organizations. Our current system provides drill-down visualizations including stacked-time-series, treemaps and sparklines.

The rest of the paper is structured as follows. Section II outlines the design and architecture of Sequencer system.
Section III discusses our results in the context of stories pertaining to the Massachusetts election results from early January, 2010. Section IV outlines the visualization capabilities of Sequencer, followed by a discussion on user generated media in Section V and conclusions in Section VI.

II. SEQUENCER DESIGN

The primary goal of the research is the creation of a scalable temporal tracking system for topic detection and named entity recognition with visualizations to enable exploration of patterns and changes in the extracted entities and their social contexts. To achieve this goal, Sequencer features a four stage process: crawl, cluster, extract, and visualize. The process is invoked on a set schedule and begins with the crawling phase. Figure 1 outlines the overall architecture of Sequencer:

![Figure 1 - The processing phases used in Sequencer.](image)

The crawling phase is handled by Apache Nutch [5], an open source web crawler built on Apache Lucene, an information retrieval framework, and Apache Hadoop, an open source implementation of Google’s MapReduce [6]. Apache Nutch is a fully distributed web crawler that allows us to handle large-scale web crawling tasks, but can also be run in a vertical search manner (such as limited to a specific news organization). Nutch was chosen due to its extensible nature and implementation of a similar scoring methodology as PageRank [7].

The output of the crawling phase is an inverted index annotated with the crawl time. This inverted index is a data structure that provides a reverse mapping of terms to content, and is designed to allow fast full text searching. In the case of Sequencer, we use an inverted index for access to a documents term frequency count for generation of the Term Frequency-Inverse Document Frequency (TF-IDF) measure.

When a new inverted index is created, agglomerative hierarchical clustering [8] is performed and then partitioned using a tuned cutoff value to estimate the optimal level of granularity in the clusters.

Clustering is performed by comparing documents in the inverted index, and calculating their vector similarity using the Cosine Similarity formula; this is a measure of similarity between two vectors of n dimensions calculated by finding the cosine of the angle between them. Using the inverted index, documents are represented by their TF-IDF vectors, and a score generated using the following:

\[
\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{||\mathbf{A}||| \cdot ||\mathbf{B}|||}
\]  

Sequencer then uses a single link approach, whereby each step merges the two clusters which have the smallest maximum pairwise distance among the set of all clusters. Inter-cluster comparison is used to detect changes in clusters between the previous crawl and the current crawl. The detection of cluster changes between each day’s crawl is critical to the operation of Sequencer and is the driver behind identification and extraction of new temporal sequences.

Cluster change-set detection is performed by comparing URL’s of the existing clusters and is placed into three distinct states. If the cluster has merely had URLs added to it that did not exist before the current crawl, the cluster is determined to be in the growing state. If the cluster has removed or added URLs that were already present, then the cluster is set to the shifting state. If the inter-cluster comparison detects a significant cluster sharing no common articles from the previous cluster results, a new topic is created and tracked in the Sequencer system.

Clusters in the shifting state undergo further processing to identify whether or not the underlying topic has changed. This is done by the calculation of an entity similarity score. The previous cluster is compared to the current cluster by extracting the named entities from both clusters and calculating the entity similarity score, which uses a score entity inverse-document-frequency, closely related to the term-frequency inverse-document-frequency [9] in information retrieval.

\[
e_{idf} = \frac{e_{i,f}}{\sum e_{i}}
\]

This leads to the calculation of an entity overlap score, which can be viewed as the change in the entity count from the previous document to the current document. If the previous or current document does not exist in the cluster,
then the score is merely the idf of the sole existing document. This calculation is necessary due to the current methodology for capturing rapidly changing events on news sites. Since multiple news sites append information onto existing documents, documents can change state from cluster to cluster.

\[
\text{overlap}_{\text{entity}}(doc_{\text{prev}}, doc_{\text{curr}}) = \text{idf}_{\text{prev}} - \text{idf}_{\text{curr}}
\]

(3)

Once the entity overlap is calculated, the overlap between two clusters can be calculated by summing the overlap for each entity. Values closer to zero represent clusters with very minimal change in the underlying topic, whereas documents with values farther from zero represent clusters that have undergone radical changes and can be deemed as entirely new topics.

\[
\text{overlap}_{\text{cluster}}(\text{cluster}_{\text{prev}}, \text{cluster}_{\text{curr}}) = \sum_{\text{entities}} \text{overlap}_{\text{entity}}
\]

(4)

Equation 2 can be viewed as a modified chi-squared test that tests the observed frequency of named entities on the current date against the expected frequency of the existing data. Chi-square is a statistical test commonly used to compare observed data with data we would expect to obtain according to a specific hypothesis. In the case of our example described below, the Massachusetts election results, the expected hypothesis is that the observed entities are within a statistical likelihood of the expected entities that have previously been identified. Representative clusters produced by the Sequences system are shown in Figure 2. We discuss our results further in the following section.

Clusters with a set of changes outstanding, or deemed as a new topic, are sent to the next phase to extract named entities from the text. Sequencer uses a maximum entropy approach [10] using models trained from existing news sites. For each URL in the cluster, named entity recognition is applied, identifying people, locations, and organizations, and summed with all other URL’s in that cluster. This creates a temporal ‘snapshot’ of the cluster, and can be compared to previous clusters in the sequence.

<table>
<thead>
<tr>
<th>Jan 16th</th>
<th>Jan 17th</th>
<th>Jan 18th</th>
<th>Jan 19th</th>
</tr>
</thead>
<tbody>
<tr>
<td>Republicans</td>
<td>4 Democrats</td>
<td>4 Martha Coakley</td>
<td>5 Scott Brown</td>
</tr>
<tr>
<td>President Obama</td>
<td>3 Democrats</td>
<td>4 Scott Brown</td>
<td>5 Martha Coakley</td>
</tr>
<tr>
<td>Scott Brown</td>
<td>2 Martha Coakley</td>
<td>2 Obama</td>
<td>4 Democrats</td>
</tr>
<tr>
<td>Martha Coakley</td>
<td>2 Scott Brown</td>
<td>4 Democrats</td>
<td>4 Obama</td>
</tr>
<tr>
<td>Democrats</td>
<td>2 Obama</td>
<td>3 Republicans</td>
<td>4 Obama</td>
</tr>
<tr>
<td>Ted Kennedy</td>
<td>1 Massachusetts</td>
<td>2 Massachusetts</td>
<td>2 Massachusetts</td>
</tr>
<tr>
<td>Massachusetts</td>
<td>1 Ted Kennedy</td>
<td>1 Ted Kennedy</td>
<td>1 Ted Kennedy</td>
</tr>
<tr>
<td>Robert Gibbs</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GOP</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>McCain</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3 - Extracted named entity data by date.

The maximum entropy approach turns the problem of recognizing names in textual elements into a statistical classification problem, where the problem is to estimate the probability that a classification, such as Person, Location, or Organization, occurs within the specified context, such as a sentence fragment. The advantage of this approach is that classification is not a binary task; rather, each classification has an associated probability that the term has that classification. This permits the system to handle ambiguous results in multiple classifications for later processing, as well as to reflect uncertainty in the process.

III. RESULTS

Our initial test data set was a 30 day snapshot of CNN.com from January 13th, 2010 to February 12th, 2010. There were a few significant events during this time span, notably, the Haitian earthquake, Massachusetts senate election, and the United States healthcare reform debate.

CNN.com’s news article generation has a few interesting aspects that must be handled for effective extraction. First, CNN.com often tracks rapidly changing stories or current events (such as the Haitian earthquake) by appending information onto existing pages and creating a running log of data. Each paragraph is time stamped, but the content of each paragraph may be substantially different. For example, an article may have details regarding an airport closure in one paragraph, followed by a discussion on cell phone reception in the next. Secondly, CNN.com has user-generated content and a special section called the “NewsPulse”, which features collections of articles and video, often video from the CNN TV news broadcasts. These aggregated pages often contain information that is wholly unrelated to each other, making clustering and extraction slightly more difficult.

CNN’s iReport section is a rich source of user generated content but was not evaluated at this stage in Sequencer’s development. However, we believe the basic information extraction process will be identical to the one described. The named entity recognizer was trained on a sample set of CNN.com articles that were manually annotated. An annotation consists of \(<\text{START}>\) and \(<\text{END}>\) tags denoting the specific block of text that represents an entity. Each entity classification, such as Person, Location, etc., has a separate training file that distinguishes specific entity types. For example, the Person training file consists of a collection of text blocks similar to:

\(<\text{START}>\) Steven\(<\text{END}>\) wants to go to San Francisco.

Whereas the Location training file might include the same sentence, but structured as:

Steven wants to go to \(<\text{START}>\) San Francisco\(<\text{END}>\).

Approximately 1,000 sentences have been annotated for three different classifications: Person, Location, and Organization. Ideally, each training set would have in excess of 15,000 annotations per training file, however due to time constraints the sample size is limited. One of the resultant outputs of Sequencer is a count of entity names by day within each cluster. Figure 3 shows the summarized Massachusetts election results broken down into entity counts per day. This is a midpoint snapshot of the overall temporal event: January 19th represents the day...
immediately preceding the election. In this situation, it becomes clear that the primary entities within the event have changed in just less than four days. The starting date, January 16th (which is the start of the test data set used), indicates the primary entities are Republicans, President Obama, Martha Coakley, Scott Brown, and Democrats. By the 19th, mentions of Scott Brown and Martha Coakley have greatly outpaced previous days’ counts.

IV. VISUALIZATIONS

These extracted sequences of named entity data lends itself well to specific types of visualizations. Currently, Sequencer is aiming towards a drill-down user interface allowing users to browse current topics and filter or search by keyword/entity.

A natural visualization for this type of data is a stacked-time-series view as shown in Figure 4. In the example, Sequencer has run on the Massachusetts senate election from mid January. This election was notable due to the nature of Democratic control of the US Senate. In this case, news coverage of the election was more focused on Democrats and Republicans than it was on the two participants, Martha Coakley and Scott Brown. However, the stacked-time-series view shows how that structure changes during the course of coverage. At the midpoint, which represents the day of the election, both Martha Coakley and Scott Brown become the primary entities within the story. By the end of the view, two days after the election, Scott Brown, the victor of the election, was the primary entity. Martha Coakley had fallen drastically off, as her role in the story had diminished.

The next visualization is a TreeMap view, typically used for displaying tree-structured data by using iteratively nested rectangles. As shown in Figure five, size and color of individual rectangles are used to convey information in this visualization. In this case, a temporal sequence is converted to a limited-depth tree structure, where the depth is defined by the “temperature” of an entity, and temperature is defined as the rate of increase in mentions per cluster. An increase in mentions tends towards red, and a decrease in mentions tends towards blue. This view provides a quick assessment feature of the primary entities within a topic – providing the Who, What, and Where of the Five W’s identified earlier.
VI. CONCLUSION

This paper introduced a novel method for mining and extracting entities and their social contexts from news sites using common text mining applications in a sequential, pipelined fashion. The Sequencer system builds temporal sequences and offers visualization tools for data analysis and exploration.

Future work on the Sequencer system will focus on extracting information from user-generated portions of news media, as well as the addition of user interface elements for browsing multiple stories, user annotation, corrections, and feedback. In addition, future work of Sequencer will focus on extracting patterns from multiple web sites, and using additional information (such as shared articles, authors, etc) to explore more complex social contexts and temporal sequences.

REFERENCES


