Advances in News and Blog Analysis with Lydia

Mikhail Bautin, Anand Mallangada, Alex Turner, Lohit Vijaya-renu, Steven Skiena
Department of Computer Science
Stony Brook University
Stony Brook, NY 11794, USA
{mbautin, amallang, aturner, lohit, skiena}@cs.sunysb.edu

ABSTRACT

Lydia is a system for online text analysis currently focusing on news, blogs and Medline abstracts. The goal of the project is to build a relational model of people, places, and things through natural language processing of the input text and the statistical analysis of entity frequencies and juxtapositions.

In this paper we describe our recent development of some new text analysis features of Lydia. This includes sentiment analysis in foreign language content, extraction of relations between entities and detection of "news storms", i.e. spikes in popularity of entities caused by a common underlying event.

1. INTRODUCTION

Periodic online publications are an extensive and readily available source of information about things and events. The Lydia project seeks to build a relational model of people, places, and other entities through natural language processing of input text and the statistical analysis of entity frequencies and juxtapositions. Our current content sources include US news, blogs and Medline abstracts. For each of these sources, we maintain a website visualizing the results of our analysis: `http://www.textmap.com` for news, `http://www.textblg.com` for blogs, and `http://www.textmed.com` for Medline abstracts.

We identify named entities in the text and track the temporal and spatial distribution of them. Our system is described in detail in [1, 5, 6, 7, 8]. Text sources are spidered daily by customized website scrapers that convert articles to a standard format and store them in an archive. Then, on a daily basis, the articles are run through a pipeline that performs part-of-speech tagging, named entity identification and categorization, geographic normalization, intra-document coreference resolution, extraction of entity descriptions and relations between entities, and per-occurrence sentiment score calculation. After that the entities are inserted into a database, and cross-document coreference resolution, juxtaposition score and per-entity sentiment score calculation take place.

The rest of this paper is organized as follows. Section 2 describes sentiment analysis in foreign language texts. Section 3 deals with our rule and dictionary based entity relation extraction systems. Section 4 describes how we determine events that cause spikes in popularity time series of entities.

.

2. PROCESSING FOREIGN LANGUAGE CONTENT

Analyzing information in multiple languages becomes important in today's global environment. There are many non-English online newspapers available, and state of the art in machine translation is already good enough to allow for meaningful mining of translated text.

One use of foreign content analysis that we are exploring is sentiment score calculation in multiple languages. This has possible marketing applications, e.g. tracking popularity of or problems with a product in foreign markets. Another application is in international political research, i.e. comparing popularity of the same political figure in different countries.

We believe extracting sentiment scores from the native language might produce more interesting and precise results than just analyzing English language sources from the same countries, which are also not always available.

2.1 Related work

Godbole, Srinivasaiah and Skiena describe in [1] the sentiment analysis method used in our Lydia system. First, sentiment scores are assigned to dictionary words starting from a small seed set of words that are clearly positive or negative and navigating through WordNet synonyms and antonyms. Then the words that have ambiguous resulting score are removed from the sentiment lexicon. After that, for a given set of input documents, sentiment scores are calculated for entities by counting positive and negative words juxtaposed with these entities.

As of the time of writing, we could not find any other ongoing research on comparative analysis of sentiments scores in different languages.

2.2 Spidering international newspapers

Our spidering programs are now capable of extracting articles from foreign language news papers. Foreign language spiders detect the encoding from the headers of HTML pages, extracts the article and converts them to UTF-8 character encoding. Currently we are spidering Spanish, Arabic, Chinese, Korean, French, Italian and German newspapers from various countries.

2.3 Machine translation

These articles are translated to English and processed using our Lydia system [6]. We ran experiments with IBM Websphere Translation Server [3], Google Translate [2] and Systrans demo [11] translation APIs. At present we translate our foreign text using IBM Websphere Translation

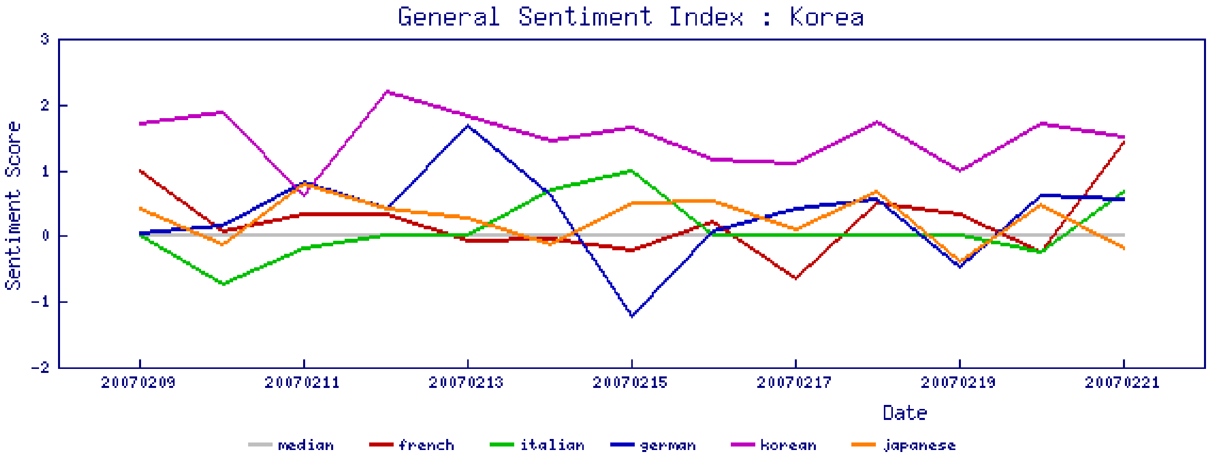

Figure 1: Sentiment index of the "Korea" entity in five different languages. "Median" here means the neutral sentiment score of zero.

Server [3, 4].

The translated text is added daily to separate databases for each source language along with newspaper and country information.

2.4 Sentiment analysis of foreign text

As we had expected, the accuracy of calculated volume and sentiment scores was directly related to the quality of translators. Specifically, if the translator uses exact translations of adjectives, this will help produce more precise sentiment scores than if the translator was just conveying the general meaning of an adjective, such as "good" or "bad".

For a given language l, an entity e and a day d we calculate entity sentiment on that day in that language as follows:

$$\frac{pos\_references_{l,e,d} - neg\_references_{l,e,d}}{num\_occurrences_{l,e,d}}, \qquad (1)$$

where $pos\_references_{l,e,d}$ and $neg\_references_{l,e,d}$ are the numbers of positive and negative words juxtaposed with entity e on day d in language l, and $num\_occurrences_{l,e,d}$ is the number of occurrences of entity e on day d.

2.5 Results

Figure 1 shows an example of sentiment index comparison of the entity "Korea" across five languages over a period of 13 days. Our sentiment analysis component of the Lydia system [1] calculates positive or negative sentiment score for each occurrence of an entity in an article and accumulates the results on daily runs. These scores when normalized to the number of occurrences for their corresponding languages reveal interesting facts. As seen from Figure 1, the overall sentiment of Korea appears to be high in Korean language newspapers as compared to others.

2.6 Future work

We are planning to normalize sentiment scores to make sentiment scores from different languages more comparable. We need this to compensate for the fact that translation quality is different for different languages, and the output of translators for some languages seems to use stronger adjectives than for the others, resulting in higher absolute value of sentiment scores. Our approach to this will be to calculate the average "strength" of sentiments in each language by averaging the absolute value of sentiment score for many entities across a certain number of days, and using these average values to normalize sentiment scores for each language.

3. ENTITY RELATION EXTRACTION

News text does not only contain information about entities, but also information about relations between them. There are multiple types of relations, e.g. family relations between people, authoring relations between persons and book/movie titles, employment relations between an employer and an employee and various kinds of other, less structured relations that can be inferred from text. This information can be very useful for applications such as question answering [5], entity graph visualization, and automated mining of semantic information from unstructured text.

3.1 Related Work

In [10], authors use a machine learning approach to mining relations between entities. They consider employer-employee, familial, person-home and organization-location relationships, and predict whether a given pair of candidate entities are in a relationship via an SVM classifier. The features used include the sequence of words in the parse tree on the path between headwords of two entities, the sequence of part-of-speech tags on the same path, full text between two entities, the ordering of two entities, bigrams on the head word path etc. Maximum F-scores of 70%-80% depending on relation category are reported.

In [9], the authors mine entity relations by matching words between entities to dictionaries. Their focus is primarily on relations between entities belonging to different countries and on doing time-series analysis of the resultant event data. Their dictionary scheme is not quite as sophisticated as ours. While they have support for keywords, they incorporate only rudimentary support for regular expressions. While this is not sufficient for our system, it serves them excellently. Their framework is at the forefront of a rule based system for analyzing international relations.

In our Lydia system, relation extraction was previously being done using a set of hand-crafted regular expressions matching entity tags and part-of-speech tags between them.

For example, a regular expression such as

$$\langle\text{entity}_1\rangle\ [,]\ \langle\text{who}|\text{which}|\text{that}\rangle\ [\langle\text{modal verb}\rangle]$$
$$[\langle\text{not}|\text{n't}\rangle]\ \langle\text{adverb}\rangle^*\ \langle\text{verb}\rangle\ \langle\text{adverb}\rangle^*$$
$$[\langle\text{prep.}|\text{subord. conj.}\rangle]\ [\langle\text{a}|\text{the}\rangle]\ \langle\text{entity}_2\rangle$$

would recognize that the entity "BitTorrent" is "based in" the entity "San Francisco" from the sentence "BitTorrent, which is based in San Francisco and has 45 employees, will face significant challenges ..." (The New York Times, 02/06/2007). This approach, with four different verb-based templates, was able to mine enough general form relations (i.e. without a pre-determined list of relation types) to drive a question answering system [5].

3.2 Approach

Unlike [10], in our case when a high volume of text is available, high recall is not a concern, because unrecognized relations could be mined from some other occurrences of the same entities. On the other hand, precision, speed and robustness is of more importance to us. Therefore we continue to use a hand-crafted regular expression rules in our Lydia relation extraction system. They have proven fast and very effective in processing gigabytes of newspaper text.

Our previous approach described above turned out to produce too many relations with unrestricted categories of which few were actually useful for our applications. For this reason we decided to enhance our relation extraction system with a scheme that would specialize in several different sets of relations, such as family relations (i.e. those between family members and relatives), political relations (such as between politicians and their parties, politicians and their constituencies etc.), company relations (recording relations such as those between the CEO and other office bearers to their companies etc.,) and various other types of relations.

Here is an example of such a rule for extracting a family relation:

$$\langle\text{entity}_1\rangle\ \langle\text{1-20 characters of text}\rangle\ \text{daughter},\ \langle\text{entity}_2\rangle$$

This rule looks for relations that contain the string "daughter ," between two entities. It would match expressions like "*Anna Nicole Smith*'s daughter, *Dannielynn* ...". The ⟨1-20 characters of text⟩ in the above regular expression puts a maximum limit of 20 characters before the start of the word "daughter". If this limit were not there, our regular expression would end up matching sentences like "DNA taken from *Anna Nicole Smith* will help prove Larry Birkhead fathered the former centerfold's 5-month-old daughter, *Dannielynn*". Entities located so far apart in a sentence are less likely to be in the relation we expect them to be in based on our regular expression. We choose length limits similar to this by hand so as to minimize false positives.

In addition to regular expressions, we also look at matches of the words between two entities to some pre-defined word lists (dictionaries) corresponding to a particular relation category. For example, the word "grandfather" is in the dictionary corresponding to the relation between grandparents and their grandkids. This enables us to trap relations such as "Maurice is the brother of *Curtis Jones*, who is the grandfather of *Karli Jones*." Here we look for the word "grandfather" in the relation text between two entities. The default behavior of the system is to stem a word before matching it to the dictionary, in the hopes of catching entities that contain slight variations of the word, but in this case in the dictionary definition file we instruct the system not to stem the word.

3.3 Results and future work

Comparing our new method of relation extraction to our old method, we can say that it is much more focused on a specific relation category. For example, for 140 MB of input text the old method produced 5060 relations and the new method gave 170 family relations which is approximately 3.4% × 5060. Manual examination of the result showed that almost all results of the new method met our quality criteria.

We are planning to further refine our relation extraction engine and optimize it for speed and efficiency in processing greater volumes of text, and rigorously evaluate it.

4. NEWS STORM DETECTION

One of the major statistics that Lydia tracks is the number of references to a given coreference set as a function of time. This has been graphed on our public website (`http://www.textmap.com`) since 2005 when the site first debuted, but there was no context provided. The data stood on its own, alongside information on juxtapositions and article names, but no correlation to actual events was made. A feature that we are developing this year is news storm detection, the ability to not just plot a graph of popularity for the last 30 days, but to detect anomalous spikes in the graph and determine what "event" caused the spikes.

4.1 Approach

The basic concept is to determine for each given day for each coreference set how much higher its popularity turns out to be above that coreference set's historical mean. We can also get a standard deviation by observing this particular coreference set's historical pattern of references to see how much fluctuation is normal for this coreference set. With these two pieces of information, we can determine how many standard deviations the actual number of references is above the mean. If this positive offset is above a certain threshold, we consider that the beginning of a spike, and thus a news storm, and track how long this trend lasts.

Once we have identified news storms, however, we need to rate their relative significance, to determine what the most important events on a given day are. To determine the significance of news storms, autocorrelation data is first generated for the coreference set database as a whole. What this means is that the strength of correlation between a given day's number of references for a coreference set as it relates to the number of references 1 day prior, 2 days prior, 3 days prior, etc. is calculated. Once we have this correlation data, we can look at the actual data for a given coreference set for the past few days and determine, based on the global correlations, what we expect the number of references to be today, if nothing external changes for that coreference set. By seeing how much higher a given coreference set's reference count is over its expected value, we can determine a score for that day and ultimately a score for the news storm as a whole.

We do more than just identify news storms by day, however. Once we have isolated a news storm and determined that it is significant, we wish to determine what "event" led to this news storm. For this, we examine the words present in the article titles for this coreference set both during and

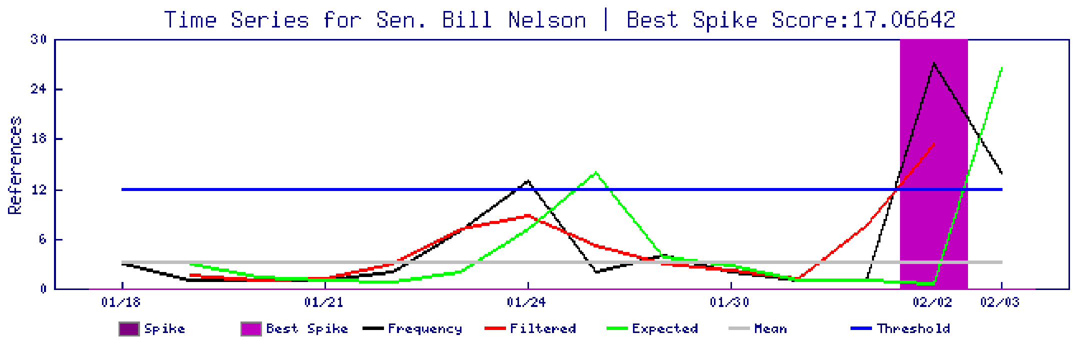

Figure 2: A sample news storm generated for the entity "Sen. Bill Nelson"

before the news storm. Using a chi-square significance test to compare each title word's occurrence before the storm relative to its occurrence during the storm, we can extract assign a score to show us the keywords that are significant within the storm, but much less significant before the storm. By assigning a score to these keywords, we can now analyze the article titles that occurred during the storm and extract the highest-scoring article title as a best-guess "title" for the underlying event that led to the news storm in the first place.

With all of this information in place, we could ultimately call out such storms on the time series graph on the entity page, providing not just an indication of the storm periods, but also a caption above showing what we would call the "title" of the storm. In this way, you could see a recent historical view of the important occurrences in the life of an entity and gain a better understanding of the context behind the data shown.

4.2 Results

Figure 2 lets you see the number-of-references values that went into determining both the existence of the spike and its relative score, including the mean and threshold, along with the actual and expected values for each day. The purple region represents the detected news storm. The most significant keywords for the spike relative to before the spike were: "extension", "rejects", "senate", "medicare", and "deadline". From these keyword scores, the title "Senate Rejects Medicare Deadline Extension" was chosen from the pool of six in-spike article titles as the most representative title for the news storm.

4.3 Future work

Our work continues on the news storm analysis before we are ready to deploy it publicly to the web site. We wish to further refine our scoring algorithms to give us a list of most significant events that more closely matches what a human editor would consider to be a day's most important news. We also ultimately hope to have different lists for different types of news such as business news and entertainment news. As of now, we are detecting spikes fairly well and are generating applicable titles, but are still having some issues rating news storms relative to each other.

Our most promising current project regarding news storms is to cluster individual news storms that involve multiple entities to form "meta-storms" when it can be determined that a particular event involved multiple people. For example, currently if one entity was to murder another, it would show up as two separate significant news storms, one for the murderer and one for the murder victim. This problem shows up especially with sports events that often refer to many entities (the Super Bowl can easily generate 10-15 news storms for the important players on each team), and can easily overweight the top storms list with duplicate storms for the same event. By merging these storms into one meta-storm, we would both eliminate these duplicates and provide a more informative experience to a user browsing our site who can now explore what other entities participated in a particular storm.

5. REFERENCES

[1] N. Godbole, M. Srinivasaiah, and S. Skiena. Large-Scale Sentiment Analysis for News and Blogs. In *Proceedings of International Conference on Weblogs and Social Media (to appear)*, Mar. 2007.

[2] Google Inc. Google Translate. `http://translate.google.com/translate_t`.

[3] IBM Corporation. WebSphere Translation Server for Multiplatforms. `http://www-306.ibm.com/software/pervasive/ws_translation_server`.

[4] IBM Corporation. Guidelines for Writing Content that Will Be Machine-Translated, Sept. 2001.

[5] J. H. Kil, L. Lloyd, and S. Skiena. Question Answering with Lydia. In *The Fourteenth Text Retrieval Conference (TREC) Proceedings*, 2005.

[6] L. Lloyd, D. Kechagias, and S. Skiena. Lydia: A system for large-scale news analysis. In *SPIRE*, pages 161–166, 2005.

[7] L. Lloyd, A. Mehler, and S. Skiena. Identifying co-referential names across large corpora. In *CPM*, pages 12–23, 2006.

[8] A. Mehler, Y. Bao, X. Li, Y. Wang, and S. Skiena. Spatial Analysis of News Sources. In *IEEE Trans. Vis. Comput. Graph.*, volume 12, pages 765–772, 2006.

[9] P. A. Schrodt, S. G. Davis, and J. L. Weddle. Political Science: KEDS-A Program for the Machine Coding of Event Data. *Social Science Computer Review*, 12(4):561, 1994.

[10] N. Singh. The Use of Syntactic Structure in Relationship Extraction. Master's thesis, MIT, 2004.

[11] SYSTRAN S.A. SYSTRANLinks translation service. `http://www.systransoft.com/index/Products/Online-Services/SYSTRANLinks`.