Computer generated news site - TIME.mk

Igor Trajkovski

Faculty of Computer Science and Information Technology,
New York University Skopje, 1000 Skopje, Macedonia
admin@time.mk
http://www.time.mk/trajkovski/

Abstract. The Internet has broken down the barriers that exist between people and information, effectively democratizing access to human knowledge. Nowhere is this more apparent than in the world of news. According to a recent survey news browsing and searching is one of the most important internet activity. The huge amount of news available on line reflects the users need for a plurality of information and opinions. We believe that more information means more choice, more freedom and ultimately more power for people. TIME.mk is our attempt to connect users with the most important current news stories. It pulls together the most reported stories on its front page so that users do not have to examine the web for up-to-date stories. In this paper we present the architecture of TIME.mk and describe all the details of its functioning.

Key words: news engine, information extraction, text similarity, clustering, classification

1 Introduction

According to a recent survey [1] made by Nielsen/NetRatings for Newspaper Association of America, news browsing and searching is one of the most important internet activity. In June 2009, there were more than 70.3 millions of active news users in U.S.. The huge amount of news available on line reflects the users need for a plurality of information and opinions. News Engines are then a direct link to fresh and unfiltered stream of information. There are many commercial news engines. Google News retrieves news information by more than 4,500 sources, organizes it in categories and automatically builds a Web page with the most important news for each category. Yahoo news runs an analogous service on more than 5,000 sources. A list of commercial news engine is given in [2].

Despite this great variety of commercial solutions, we found just few academic papers on this subject. NewsInEssence [3] is a system for finding and summarizing clusters of related news articles. Chung [4] proposes a topic mining framework for news data stream. Henzinger [5] finds news articles on the web that are relevant to TV news currently being broadcast. Reis [6] proposes a tool to automatically extracting news from Web sites. We think that the few scientific publications cause the news engine technology to remain largely a secret art.

2 TIME.mk as a news engine

In this paper, we introduce a general framework to build a News Engine. Our system, TIME.mk, is a complete news engine for retrieving, indexing, clustering, classifying, ranking and delivering news information extracted both form the Web and from news feeds. Currently TIME.mk collect news from 50 Macedonian news sources (TV stations, newspapers, news agencies, etc. and every day we get more requests for inclusion), groups similar articles together and displays them in order of their calculated importance.

By linking people to so many news sources, TIME.mk makes it much easier for them to read about stories from different angles and to search for more information on issues of their interest. The strength of this project is the respect of copyright. TIME.mk never shows more than the headline, a snippet and a thumbnail image of the news article. If people want to read the entire news story or see the full photo, they have to click through to the news source's website.

TIME.mk is similar by function to the Google's search/news engine in that it collects all the news it can find, creates an index of that information, and serves it to the users via a simple web interface. Our goal is to give users the most relevant, objective results, which is why we generate them automatically and without human intervention.

TIME.mk organizes articles so that many texts from different sources of a single news story appear in a group. We call these 'news clusters'. This approach groups headlines from different publications together, providing users with multiple viewpoints on a given news story. Publishers often ask us how we decide which clusters and type of news appears on the TIME.mk homepage. The short answer is: we don't decide.

The headlines on the TIME.mk homepage are selected entirely by an algorithm, based on many factors including how often and on which sites a news story appears elsewhere on the web. Basically, we look at the number of original articles being produced and published by editors in order to determine the size of a news story, which we also weight based on how recent it is.

Say, for example, that TIME.mk registers that in a one-hour period of time a cluster of two news articles about a football match in Germany appears; whereas a news story about EU-Serbia-Kosovo talks nets 20 articles. The algorithm detects that the latter is a bigger news story and gives that cluster priority in the ranking.
The system is made up by the modules depicted in Fig. 1 and has a Web interface at `http://www.time.mk/`. In the following, we describe them.

2.1 Retrievers

The task of this module is gathering links of news articles and extracting text and images from the web pages of these news articles.

TIME.mk uses two methods for links gathering. One is with RSS and another with regular expression(s) directly from the web pages of the news sources. In

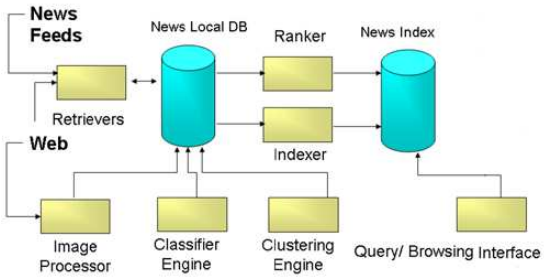

Fig. 1. Architecture of TIME.mk

the second method, for each news source we have to write regular expression(s) for detecting the relevant links.

Text extraction is the problem of selecting text and image on a web page that represents the content of the news. TIME.mk uses two methods for solving this problem. The first method sees news artislec's web page as simple array of characters, which is translated into array of numbers, one number for each character. If the character is a part of a HTML tag, then it is translated into a negative number, othervise it is translated into a positive number. Then the problem of text extraction is transformed into the search of continuous subarray with maximal subsum. The second method first creates DOM (Document Object Model) tree of the HTML data. Then it extracts all the text that is found inside news source specific set of tags (nodes).

Method	Advantages	Disadvantages
Maximal subsum	General, it works for all news sources	not suittable for short texts, it can extract text that is not part of the news
DOM tree	Accurate	requires news site specific html tags

Table 1. Comparision of the text extraction methods

If links of the news articles are not gathered by RSS, for each source we need to provide urls where links of the news articles can be found. These urls are called 'hubs'. By providing the hubs for given news source, we also provide the category of the news articles found at these hubs. Thus, the news articles enter the system as already classified (by category) and can be used as clean data for training the classifier that is used for classifying news articles that are not categorized.

2.2 Keywords extraction

Keyword extraction process is used after the text of the news articles is extracted and its task is to extract single words (terms) characteristic for the news article. Extracted keywords are used for classification and clustering of the news articles.

For solving this task we used vector space model [7] for representing news articles. A document is represented as a vector and each coordinate of the vector corresponds to a separate term. If a term occurs in the document, its value in the vector is non-zero. Several different ways of computing these values, also known as term weights, have been developed [7]. The dimensionality of the vectors is the number of words in the vocabulary (the number of distinct words occurring in all news articles).

In the classic vector space model proposed in [7] the term specific weights in the document vectors are products of local and global parameters. The model is known as Term Frequency-Inverse Document Frequency (TF-IDF) model. The weight vector for document d is:

$$v_d = [w_{1,d}, w_{2,d}, ..., w_{K,d}]^T$$

where:

$$w_{t,d} = tf_t \cdot \log \frac{|D|}{|D_t|}$$

and:

- tf_t is term frequency of term t in document d (a local parameter)
- $\log \frac{|D|}{|D_t|}$ is inverse document frequency (a global parameter), where $|D|$ is the total number of news articles; $|D_t|$ is the number of news articles containing the term t and K is the number of distinct words occurring in all news articles.

After the term weights are calculated the news article is represented with the top N_{kw} terms (keywords) and its appropriate weights. At the end this vector is normalized ($\|v_d\| = 1$).

One applications of this kind of representation is the computation of news articles similarity, by calculating the angle between news articles vectors. In practice, it is easier to calculate the cosine of the angle between the vectors instead of the angle, that is why it is called *cosine similarity*:

$$\cos\theta = \frac{v_1 \cdot v_2}{\|v_1\|\|v_2\|}$$

A cosine value of zero means that the news articles are orthogonal and have no major similarities.

2.3 Image processor

This module analyzes information stored in the news database and tries to enrich each news article with an associated image. In the easy case the news source has already associated an image to a given news in the RSS feed. In other cases, we have a news n, extracted by a Web page p or by a RSS feed, with no associated image which refers to a Web page p. In this case we use heuristic to identify the most suitable image to be associated to n, from other news articles that have an image. We take the image from the n's most similar news article (that has image) in the news database.

2.4 Classifier engine

All not classified news articles needs to be classified. The categories in the system are predefined and given in Table 2. Note that a (relatively large) part of the RSS feeds and hub pages are already classified from the originating news source. As a consequence, the key idea for classifying is to use the classifier in a mixed mode: as soon as an already classified news by a news source is seen, the classifier is switched in training mode; the remaining unclassified news are categorized with the classifier in categorizing mode. We use a naive Bayes classifier [8]. The probability of a given document d composed of words $w_1, w_2, ..., w_n$, belonging to a class C, is:

$$p(C|w_1, w_2, ..., w_n) = \frac{p(C)\ p(w_1, w_2, ..., w_n|C)}{p(w_1, w_2, ..., w_n)}$$

In practice we are only interested in the numerator of that fraction, since the denominator does not depend on C and the values of the words w_i are given, so that the denominator is effectively constant. Using the "naive" conditional independence of occurance of the words w_i, the numerator is equivalent to the joint probability model:

$$p(C, w_1, w_2, ..., w_n) = p(C) \prod_i^n p(w_i|C)$$

where: $p(C)$ is the fraction of article belonging to C, and $p(w_i|C)$ is the probability that document beloning to C will contain the word w_i. Both parameters are easily computed.
Finally, the corresponding classifier is the function *classify* defined as follows:

$$\mathit{classify}(d) = \mathit{classify}(w_1, w_2, ..., w_n) = \mathit{argmax}_c\ p(C = c) \prod_i^n p(w_i|C)$$

2.5 Clustering engine

Clustering engine has a task to group together news articles that report about the same event. These groups are called news stories. We don't know in advance the exact number of news stories and their coverage (number of articles covering a single story). Therefore we use hierarchical agglomerative clustering (HAC) for solving this task [8]. HAC treat each news article as a singleton cluster at the beggining and then successively merge (or *agglomerate*) pairs of most similar clusters until all clusters have been merged into a single cluster that contains all news articles. A HAC clustering is typically visualized as a dendrogram as shown in Fig. 2.

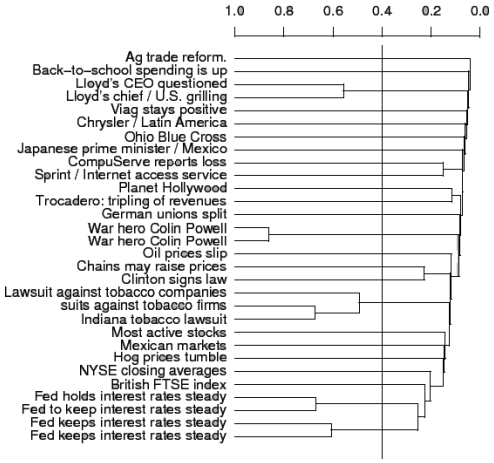

Fig. 2. A dendrogram of a HAC clustering of 30 documents is shown. Cut of the dendrogram at 0.4 is shown. 24 clusters are formed. Borrowed from [8].

HAC does not require a prespecified number of clusters. However, we want a partition of disjoint clusters. In this case, the hierarchy needs to be cut at some point. We cut at a prespecified level of similarity. For example, we cut the dendrogram at 0.4 if we want clusters with a minimum similarity of 0.4.

2.5.1 Time complexity of HAC The naive implementation of HAC is $O(n^3)$. We need $O(n)$ steps of selecting the most similar clusters, mergin these two clusters and recalculating the similarity of the new cluster with the remaining ones. Naive selection of the most similar clusters is $O(n^2)$. We used centroid clustering for calculating the similarity between clusters which give us $O(1)$ time for merging two clusters. All these steps in total give $O(n^3)$ complexity, which is quite slow if we try to cluster thousands of news articles. But if we use priority queue (PQ) for storing the similarity matrix then the total complexity will be $O(n^2 \log(n))$. Keep in mind that inserting and deleting an element in PQ is $O(\log(n))$ and selecting the minimum is $O(1)$.

2.5.2 Classification of clusters After the news stories are created as a result of the news article clustering, we need to classify each news story into one of the ten categories. This classification is needed becouse we can not expect that all articles in one cluster will be from one category. For example, event about the world economic crisis one source can publish it in the world section, and another source can publish it in economy section. We solve this problem by simple majority voting. If two categories have the same number of votes we clasify the article with the more specific category (Balkan is more specific than World, Economy is more specific than Macedonia and Word, etc.).

2.6 Ranker

Ranking news articles is a rather different task than ranking Web pages. From one side, we can expect a smaller amount of spam since news stories come from controlled sources. When a news article is issued, we can have two different scenarios: the news article can be completely independent on the already published stories, or can be aggregated to a (set of) news articles previously posted. Anyway, we stress that, by definition, a news article is a fresh piece of information. For this reason, when a news article is posted there is almost no HTML link pointing to it. Therefore, HTML link based analysis techniques, such as PageRank, can produce a limited benefit for news ranking.
Any ranking algorithm for news stories (clusters) should have at least the following four properties:

- *Time awareness*. The importance of a piece of news changes over the time. We are dealing with a stream of information where a fresh news story should be considered more important than an old one.
- *Important News articles are Clustered*. An important news story is probably (partially) replicated by many sources. From the news engine point of view, this means that the (weighted) size of the cluster is a measure of its importance.
- *Authority of the sources*. The algorithm should be able to assign different importance to different news sources according to the importance of the news articles they produce. So that, a piece of news coming from "BBC" can be more authoritative than a similar article coming from say "Kirilica", since "BBC" is known for producing good stories.
- *Diversity*. News story reported by big number of sources should be more important than news story reported by small number of sources.

This is the formula used by TIME.mk for calculating the weight of the cluster c at moment t:

$$WC(c,t) = SourceEntropy(c) \cdot \sum_{i=1}^{k} WN(n_i,t)$$

where:

- k is the size of the cluster c. n_i, $(1 <= i <= k)$ are the news articles composing the cluster c.
- $SourceEntropy(c)$ represents the entropy of the set of news sources that are included in the cluster. This value is scaled and its value range is from 1 to 2. If all sources in the cluster are different $SourceEntropy(c) = 2$. If $k > 1$ and all sources in the cluster are the same then $SourceEntropy(c) = 1$.
- $WN(n_i, t)$ the weigth of news article n_i at time t which has been published at time t_i:

$$WN(n_i, t) = A(source(n_i)) \cdot e^{-\alpha(t-t_i)}, \quad t > t_i$$

  The value α, which accounts for the decay of "freshness" of the news story, is obtained from the half-life decay time θ, that is the time required by the rank to halfe its value, with the relation $e^{-\alpha\theta} = \frac{1}{2}$. This α can depend on the category to which the news article belongs. For instance, it is usually a good idea to consider 'Sport' news decaying more rapidly than 'Fun/Showbiz' news.
- $A(s)$ accounts for the authority of the source. One source is more authoritive if it is more cited than other sources. Every time when our system detects a duplicate of a news article, published by two or more different sources, the source that first published the article gets a credit. Duplicate news articles are detected by n-gram method.

The final step is sorting the clusters according to their weight and creation of the static HTML files that are served to the users.

3 Implementation details

The news engine is running on a single PC with a Pentium IV 3GHz, 2GB of main memory. It is completly implemented in programming language Python.

Currently, we selected a list of 50 news sources (most popular TV stations, newspapers, news agencies, etc.). For efficiency reason, the space of news sources is partitioned and this module is composed by several processes which run in parallel. The data is collected 24h per day, updated every 10 minutes - 144 times a day, and the stream of information is stored into a news database.

The complete cycle of crawling the all 50 sources, extrcting the text from the new news articles, clustering the news articles and scoring the news articles and news stories, takes approximatelly 3 minutes (depending on the response time of the web sites of the news sources). Clustering is performed only on news aritcles published in the last 2 days, which gives us approximatelly 2500 news articles. This number of news articles is clustered in less than one minute.

Number of keywords, N_{kw}, was determined experimentaly and it has a value of 7. We don't take all the words in a news article in its representation, becouse most of them have small weight compared to the 5-6 most important keywords. Also clustering algorithm is running faster becouse the time complexity of the cluster similarity function is linear in the dimensionality of the representation

vectors. We can not chose less keywords, becouse small number of keywords can not accurately represent the content of the news article.

Also, not all news sources have the same categories as those predefined in our system. Therefore we need to map news source's categories into TIME.mk categories, if such a mapping is possible. Othervise news articles are left uncategorized and we leave classifier to categorize them. In some cases we help the classifier, by suggesting that news article is belonging into one of the two presumed classes, so classifier need to decide between two classes, not between ten classes. For example, if some source has not category Balkan, it publishes its article about Balkan in section World. We suggest to the classifier that news articles from this hub belong to the category Balkan or World. Now classifier has much easier job to do (the classification error is smaller).

4 Results

For space reason, we report just the most important results. For evaluating the quality of the results, we used the data set collected by TIME.mk, gathering news articles from more than 50 continuously updated sources. The data set consists of about 300,000 news articles collected over a period of one year (from 01/07/08 to 30/06/09) and classified in 10 different categories (see Table. 2).

Table 3 and Table 4 present the precision and recall data of the two classification problems. In the first problem, if the news articles was labeled with two categories, the classification was considered correct if classifier classified the article in one of these two categories. For the second problem we used randomlly choosed 100 (per category) manually labeled clusters as an evaluation set. Selected clusters had minimum two news articles. As we can see the precision and recall numbers are hight, becouse the probability that more than half of the articles to be incorrectly classified is very low. Most of the rare errors, come from classification of small clusters (size two or three) that rearly come on the homepage, so basically on the homepage we have more than 99% accuracy.

Category	#news	**Category**	#news
Macedonia	58,596	Sport	43,498
Balkan	19,341	Chronicle	10,108
World	29,647	Culture	16,933
Economy	29,754	Technology	5,755
Skopje	7,325	Fun/Showbiz	37,798
Uncategorized	58205	**Total**	316960

Table 2. Number of news articles, per category, in a data set.

Category	Precision	Recall	**Category**	Precision	Recall
Macedonia	84%	90%	Sport	92%	86%
Balkan	82%	81%	Chronicle	80%	77%
World	90%	87%	Culture	78%	82%
Economy	78%	80%	Technology	91%	94%
Skopje	73%	62%	Fun/Showbiz	89%	88%

Table 3. Precision/Recall of the Naive Bayes classifier for classifying articles.

Category	Precision	Recall	**Category**	Precision	Recall
Macedonia	95%	98%	Sport	98%	96%
Balkan	94%	96%	Chronicle	98%	98%
World	98%	98%	Culture	98%	96%
Economy	97%	96%	Technology	98%	98%
Skopje	98%	96%	Fun/Showbiz	98%	97%

Table 4. Precision/Recall of the majority voting algorithm for classifying clusters.

5 Conclusion and future work

In this paper we have presented an implementation details of a full scale news engine. Our work has been motivated by the large usage of news engines versus the lack of academic papers in this area. An extensive testing on more than 300,000 news articles, posted by 50 sources over one year, has been performed, showing very encouraging results.

The future work on the engine will be focused on scoring the clasters. At the moment the score of the news articles does not depent from the amount of new information that they introduce. For example if new news article is published and is clustered in a news story, most probably it will be ranked first in its cluster (except if it is duplicate), even if it does not introduce any new information. The idea is to include amount of new facts, that one article brings to the news story, into the score of the news article. This will require a more advanced semantic model for representing news, compared to the current "bag of words" model.

References

1. http://www.naa.org/PressCenter/SearchPressReleases/2009/NEWSPAPER-WEB-SITES-ATTRACT-MORE-THAN-70-MILLION-VISITORS.aspx
2. http://searchenginewatch.com/
3. D. Radev et al.: NewsInEssence - a system for domain-independent, real-time news clustering and multi-document summarization, Proc. of the First Int. Conf. on HLT Research, p.1-4, March 18-21, San Diego (2001).
4. S. Chung and D. McLeod: Dynamic topic mining from news stream data (2003).
5. M. Henzinger et al.: Query-Free News Search, In Proceedings of the 12th International WWW Conference (2003).
6. D. Reis et al.: Automatic Web News Extraction Using Tree Edit Distance, In Proc. of 13th WWW Conference (2004).
7. G. Salton et al.: A Vector Space Model for Automatic Indexing, Communications of the ACM, vol. 18, nr. 11, pages 613-620 (1975)
8. C. Manning et al.: Introduction to Information Retrieval, Cambridge Press. (2008)