Extracting Hot Events from News Feeds, Visualization, and Insights

Zhen Huang and Alfonso F. Cardenas, University of California, Los Angeles

Abstract— **Our aim is a real time news extraction system capable of identifying up-to-date "hot news" from large amounts of news reports on the internet. We show two important characteristics of hotness: short-term burst and long-term historical variation. Based on these factors, we define our problem: given all news articles on the current day *p* and previous news archives, identify the hot news on day *p*. Our novel system consists of four parts. First, all entities (atomic elements in text such as the names of persons, organizations, locations, etc) which occurred on the current day are extracted. Second, hot terms on day *p* are identified with our novel term weighting scheme for hotness. Third, each news article on the current day will be represented as a vector of hot terms with frequency. The last step is to cluster all new articles on day *p* and rank the result clusters with their hotness score. The scheme is also adaptive since the hot terms extracted are different from day to day, and reflect the hot events on each day. The focus of this paper is to illustrate the visualization and the insight of the hot-news through time, not the underlying algorithm details we developed.**

Our experiments conducted on a Yahoo! News dataset indicate that our term weighting scheme incorporating the two important features of hotness evaluate the significance of each extracted entity properly.

Index Terms— **Information retrieval, visualization**

I. Introduction

There are different kinds of news from our daily newspaper and news agents. A large portion of news articles are about the topic that constantly appear, almost every day, like Iraq and Congress. Some of the news articles were only interesting to a small amount of people. They are there just because of the integrity of news. There are much fewer reports about it.

We observe that there may be a burst of news articles about an event that rarely or never has been mentioned before. Furthermore, an event cannot be hot forever. Some hot news might be hot for only one day and submerged in the sea of information the next day, while some others might last for several days. Some hot topics never disappear and become daily topics since their appearance, like "*iphone*" and "*stimulus plan of the current economy*". Hot events which last for several days would be considered hot on the first few days they appear and less hot the following days with the decay of their hotness and emergence of other hot events. In a streaming environment, the burst size and different stages of burst need to be considered for hotness.

Fig. 1 gives the definitions of terms we use. The hotness of a certain event will decay along with the elapse of time. An article can be estimated by a set of certain entities shown in the article. In order to examine the burst of events, we estimate it by examining the burst of certain terms. The burst of a term is related to the number of documents in which it occurs and the time length of being popular. The historical variation of a term indicates how often this term appeared in history. By combining these two factors together, we can compute the hotness scores for all entities which occur on the current day and get a list of hot terms for that day. With the hot terms, we cluster the news articles which are featured by these hot terms and rank the result clusters by their hotness score.

Hotness definition

There are many definitions of the concept "hotness" [3] [11] [12]. One of the intuitive ideas is that hot events are the events on which a lot of news articles are written and thus we only need to cluster the news articles on the current day and choose the top clusters in size as the hot events. However, it is not necessarily the case that the larger a cluster is, the hotter that subject will be. We have to take the short-term burst and the historical variation into consideration. For example, there were 24 news articles about Iraq on Jan 04, 2007, which is the largest cluster that day. However, it was not the hottest subject because there was a lot of news about Iraq in history constantly.

We define "hotness": a burst of the number of articles related to an event during a short time span, wherein the event did not occur often (historical variation) in history. As discussed in Allan et al. [4], event is defined as "some unique thing that happens at some point in time". News articles about an event could be anything related to this event. It could be news reporting this event from different sides. There is a clear distinction between an event and a topic. As shown in Fig. 1, topic is defined as a broader category of events, or a class of events. "Volcanic eruption" is a topic, while "the eruption of Mount Pinatubo on June 15th, 1991" is an event [4].

Document clustering

There are many methods proposed for document clustering [5]. HAC (Hierarchical Agglomerative Clustering) [5] is one

Alfonso F. Cardenas, Author, Computer Science Department, University of California, Los Angeles, Ca 90024 (email: cardenas@cs.ucla.edu)

Author Zhen Huang, student Computer Science Department, University of California, Los Angeles, Ca 90024 (email: zhenhuang@ucla.edu)

of the hierarchical methods and starts with each article being placed into its own cluster and greedily chooses the closest pair of clusters to merge one at a time. To select the closest pair of clusters, there are many similarity comparison methods, such as Dice, Jaccard, Overlap, Cosine, and L1 Norm. Although HAC can model arbitrary shapes and different sizes of clusters, it cannot be used directly for news stream applications due to the following two reasons [1]. First, HAC builds a dendrogram which then relies on human beings to partition it to produce actual clusters. This step is usually done by human visual inspection, which is a time-consuming and subjective process. Second, the computational complexity of HAC is expensive since pair-wise similarities between clusters need to be computed.

Due to the frequent update and streaming nature of a news source, all text mining methods on news streams should be updatable. Much research [2] [3] has been focused on identifying hot news through exploring news articles from different news sources. Their case is different from our case because many other characteristics could be used by them, such as news source reputation. The data set we use is from the Yahoo! News RSS feed. It serves as an aggregator of the news from different sources. All the news articles are mixed together and their origins are unknown.

Article representation

There are several ways to represent a document. For example, a document could be represented as a bag of words which appear in the article or a vector of distinct terms with their weight (significance). The sequence of words and the structure of the text such as sentences and graphs are ignored. Because of different purposes, different words are extracted from the source article. For *tf-idf (term frequency-inversed document frequency)* which is used in the document similarity measurement, all terms that appear in the article need to be considered. In the news environment, entities such as persons, organizations, and places play an important role in identifying an event or news article in our case. “Hot” terms are entities that are contained in news articles about a hot event. To weigh how hot an event is, we have to weigh how hot the entities about the event in the news article are. There are many ways to evaluate the significance of a term. A detailed description of our term weighting scheme is presented in Section 4.

II. System Overview

The goal of our system is this: given all news articles on the current day *p* and previous news archives, identify the hot news on day *p*. Fig. 2 shows the overview of our system. The raw data is all news articles on the current day *p*. After preprocessing, each news article is represented by a vector of words with their term frequency in this article. The preprocessing includes stop word removal and named entity extraction. With the entities extracted from each article on day p and their historical occurrence extracted from the historical news article achieve, we identify the hot terms on day p with our novel term weighting scheme for hotness. The historical occurrence of the entities is needed here because they are used to compute the stage of the burst and the historical variation, which are the two key characteristics of hot terms by our hotness definition.

Since a large portion of terms are not hot terms, only terms with a hotness score above a certain threshold are extracted and the results are put in a list called the “traceable list”. With the “traceable list” and their hotness score, we represent each news article on day *p* by a vector of hot terms which occurs in that articles and is in the “traceable list”. Since only the terms in the traceable list are considered, the dimensions of the feature vector of each news article are greatly reduced and the consumed computation resource is highly reduced.

With each news article on day *p* represented by its feature vector, the hierarchical agglomerative clustering (HAC) method is employed to group the articles into clusters. Since each cluster contains all news articles about an event and related events, it is also given a hotness score to show how hot this event is. The highest hotness score of the terms in the news articles in the cluster is considered as the hotness score of the cluster. With the hotness score given to each cluster, we can rank all clusters with the score from highest to lowest. At this stage, we can answer the user’s question: “what is the hot news today?”

Furthermore, our system works in a real-time fashion with one day as the basic time unit. The hot terms and hot new articles extracted are different from day to day, and reflect the hot events on each day.

III. Term Weighting Scheme

After preprocessing in our system, each news article is represented by a vector of entities with their term frequency. The next step of our system is to derive a brand new term weighting scheme for the hot term identification from all terms which occur on the current day. There are many term weighting schemes proposed for different purposes, such as *TF-IDF* [6], *TF-ISF* [7], *TF-PDF* [8], etc.

We introduce a new term weighing approach with the following heuristics:

- The method should be *updateable*. Hundreds of news is published by single news agents every day. With the growth of news articles, it is essential to develop incremental methods to extract hot events or hot topics. The dataset is not divided into a training set and a test set. The historical statistics should be updated adaptively. A hot event does not remain hot forever, but only during certain time periods.
- Hot event should be detected early and be adaptable to the life cycle of an event
- Parameters do not need to be manually set
- Consuming minimal computation resources

Instead of consuming a lot of computation resources like Topic Similarity Measurement and document clustering, we provide an efficient and fast-adaptive approach that we shall not detail in this paper. [3] is only interested in extracting “hot topics” from a given set of text-based news documents published during a given time period. Our approach can work in a streaming fashion. Furthermore, instead of identifying targeted news article as hot or not hot [3], we give it a hotness score.

We introduce our two critical properties of a hot term: “short term burst” and “long term variation (rareness)”. In news streams, newly shown terms have significant importance. For instance, a unique term in a news stream may imply a new technology or event that has not been mentioned in previous articles. For each named entity, we compute the short-term score. This is considered to be the burst score. We also compute the historical variation score, which is directly related to how often this term occurred in history. These two scores are combined then to get the actual hotness score for this term.

Our approach examines all factors related to the definition of hotness and follows the following rules:

1. Important (hot) terms appear more frequently within a document than unimportant terms do.
2. From the view of a short term period, the more times a term occurs in all news articles, the stronger its discriminating power becomes. A burst of certain terms is a strong indicator of an event.
3. From the view of a long term period, the less variation a term has throughout history, the weaker its discriminating power becomes.

In this way, terms like “*Iraq*”, “*U.S.*” should get a lower score. Another example is the term “*iphone*”. When the term “*iphone*” was first introduced on Jan 09, 2007, it should have a high score. As time goes on, there are many news articles about “*iphone*” every day. The term “*iphone*” became more pervasive and has less variation of its occurrence, and thus it is seen as less hot.

Short-term score (burst score) computation

The number of term appearances in each document is not considered as a factor of term hotness. No matter how many times this term appears in a document, it is just in one article. The more times this term appears in one article, it is highly possible that this term is the “topical” term for this article. However, it doesn’t mean this term is hot. To measure the burst of a term, it is more reasonable to consider the short-term document frequency over the days of certain window size, which is the total number of documents containing this term each day. The total number of documents each day is more important than the term frequency in one new article when judging the hotness of a term. For example, if the term “Steve Jobs” appears many times in one news article, this frequency does not make this term more hot if “Steve Jobs” only appears in that article on that day. For each entity, its short term score is computed with a formula we proposed base on this rationale:

1. The more news articles contain this term (document frequency) on that day, the hotter this term is on that day.
2. If two terms have the same score value, the tie is broken by the number of news articles containing the term on day *(p-1)*. News articles containing the term on previous days (of *window_size*) should also be taken into consideration. This is natural for an event in a news article. Some events last only one day, while some others last more than one day.
3. The more days before time point *p*, the less contribution it will make towards the hotness score of the same event on day *p*. The document frequency of term *t* on day *p* should have more weight than the document frequency of the same term on day *(p-1)*. Viewers will lose interest in the same event eventually, so the hotness may decay exponentially with the elapse of time.

The window size is set to four days based since through observation we saw that news articles related to the same event usually do not last longer than three days. We also assume that the articles on the previous four days have a positive effect on hotness, while documents further back have a negative effect on the hotness. The longer an event lasts, the less hot the event would be.

In our approach, one day is considered as a basic time unit when the document frequency is measured, which is natural for news articles. The update rate of news articles in the news streams tends to be one day in general.

Historical variation score computation

In order to measure the hotness of a term, its long term variation also needs to be considered, which is denoted as *hist_vari(t, p)*. A large variation shows that the term was more burst-related, rather than terms like “*U.S.*” which appears in our news articles every day or every now often. A term with large variation denotes that there was a burst of this term (event) in history. The variation could be the number of days a term occurred or the number of documents this term appeared in. We hypothesize that a larger variation leads to a hotter term. There are several factors which might be related to its historical variation that we examined. We introduce the following concepts, definitions, and formulations elsewhere to compute the historical variation score. Examples of the needed visualizations for these are below.

Hotness score computation

As mentioned earlier, each news article is represented by a vector of entities which contain information about people, locations, organizations, etc. First, we obtain a list of terms occurring on day *p*. For each term *t*, we obtain its short term score (burst score) and historical variation score with the above mentioned methods. The next step is to compute the hotness score of all terms that occurred on day *p*, formally *hotness_score(t, p)*. As discussed above, the greater the burst score, the larger the burst size, the hotter the term. Similarly, the larger historical variation the term has, the less common the term, and the hotter the term. Thus, the hotness score of term *t* is computed as:

$$hotness\_score(t,p) = burst\_score(t,p) \cdot historical\_variation(t,p)$$

With the hotness score of each term computed, the terms which occur on day *p* are ordered from highest to lowest by their hotness score. Terms with a hotness score above certain threshold are extracted and the list is called the "traceable list".

Our term weighting scheme is updatable and thus adaptive. The "traceable lists" are different from day to day. For a certain term, it may be hot on some day with the burst of an event, while not as hot on some other days with the decay of this burst. Our weighting scheme reflects the life cycle of a burst and the fast-paced news environment, in which fresh news emerges every day. Our scheme also reflects the historical variation of a term. More common terms like "*U.S.*", "*Iraq*", and "*California*" appear less in our "traceable list", except when there is really a burst of these terms. An example is when President Bush delivered a speech on Iraq which drew global attention on Jan 09, 2007. This case can also be modeled correctly in our scheme.

IV. News Article Clustering

Since the hottest terms are extracted each day, the next step is to identify the hot articles from all news articles on day *p*. During the first phase, each article on day *p* is represented as a vector of terms and their term frequency. We represent each article in a different way: a vector of terms which were in the article and also in day *p*'s traceable list. Term weight is given by the following equation: *term_weight(t,d,p) = hotness_score(t,p) normalized_term_frequency(t,d)*

term_weight(t, d, p) is the term weight of term *t* in document *d* on day *p*; h*otness_score(t, p)* is the hotness score of term t on day p; *normalized_term_frequency(t, d)* is normalized term frequency of term *t* in document *p*. The term frequency of term *t* is taken into consideration because terms which occurred more frequently in an article tend to be topical terms for this article. The more a term appears in an article, the more important this term is to the whole article. Fig. 5 shows the representation of an article.

The next step is to cluster the articles. The vector is represented as a feature vector for this article for clustering.

The Hierarchical agglomerative clustering method is used to group articles into clusters of same events. The cosine similarity measure is used on as our similarity measurement. The next step is to rank the result clusters. In order to do that, each cluster has a set of terms which appear both in day *p*'s traceable list and the articles of this cluster. At first, we try to give the hotness score of a cluster as the average hotness score of all terms that appear in both the traceable list and the cluster. The highest hotness score of the term in the set is considered as the hotness score of the cluster to account for clusters that have very few but very hot terms. Formally, the hotness score of a cluster is given by:

$$\Re = \{terms\ appears\ in\ the\ traceable\ list\ and\ the\ articles\ in\ cluster\ c\}$$

$$hotness\_score(c,p) = \max\{hotness\_score(t_i,p) \mid t_i \in \Re\}$$

V. Hot Events Experiments and Visualization

In order to evaluate the accuracy rate of the hot news articles generated, we implemented a real-time hot subject extraction and rank system in Java. The system pipeline is implemented with the IBM UIMA framework [9]. We evaluated our system against the Yahoo! data sets. LingPipe [10] is used for entities detection (e.g. people, organization and location) from each news article.

We explore the effectiveness of our approach to identify and rank "hot" entities and consequently "hot" subjects for the given date, e.g., what are the hot terms and hot subjects on a certain date, such as Jan 04, 2007. Given the date Jan 04, 2007, only the news articles on and before that day are available to us.

We first examine the set of entities generated from an article to see whether the set covers most entities shown in the article such as people, locations, and organizations involved. Then, the hotness score of each term on day p is computed. With the hotness scores, we get the hot terms on that day and cluster the news articles, which are represented by the extracted hot terms. By comparing with manually tagged hot news articles, we get the accuracy recall rate of our algorithm.

Dataset description and overview of experiments

The Yahoo! news dataset is a collection of 125,871 news articles from all Yahoo! News RSS feeds, collected between 01/2006 and 03/2007. Through LingPipe, which implements a dictionary-based chunker that performs approximate matching, 317,475 distinct entities are recognized and extracted. Other words and the order of each word are not considered. LinePipe extracts entities effectively. Most of event-related entities are extracted. Fig. 6 shows the top 50 entities with highest term appearance in our dataset. We filtered out 274,192 infrequently occurring entities by requiring that an entity should occur twice in an article, leaving 43,284 entities in the dataset.

In order to fully validate our system, we conducted three experiments. First, we compared hot terms extracted by our approach with Term Frequency, Documents Frequency, TF-PDF and the weighting scheme described in [2]. This experiment validates the effectiveness of our method over these weighting schemes. In the second experiment, we demonstrate how we can identify and rank real-time hot terms effectively.

Experiment 1

Figs. 7 - 9 show the top 50 entities on Jan 04, 2007 in order from high to low by term frequency, document frequency and TF-PDF value respectively. Not surprisingly, most of the top terms are relatively common terms, such as *"U.S.", "Iraq", "Microsoft", "California"*, etc. These terms appear in news articles every day. There are two categories of these terms. One is the kind of term on which many news articles are produced, like "*Microsoft", "Google", "Saddam", "Bush"*. There are news articles on these terms almost every day.

Meanwhile, the other kind is more like containers, such as *"U.S.", "China", "California", "Los Angeles".* Events happening at these places tend to have these terms in the articles. Both kinds of these terms cannot be considered as hot terms.

Fig. 10 presents the top 50 ordered terms ordered by our weighting scheme. The result is much better than that derived by the other approaches shown in Figs. 7 – 9 and ref [2]. The relatively common terms are given low scores and none of them appear in our top 50 list. The genuinely hot terms extracted from hot articles rated by Yahoo! rank higher in our ordered list, such as *"ironport", "nardelli",* etc.

Experiment 2

Fig. 11 shows the "hotness" score change over time for the hottest terms "*iraq*", "*ironport*", and "*iphone*". "*iraq*" had a relatively low score over time since it is a common term and not considered as a "hot" topical term in our measurement. However, it gained a relatively higher score on Jan 09, 2007. The reason is that President Bush delivered an important speech on Iraq which drew a lot of attention and resulted in a lot of news articles about this topic.

Summary of results and comparison to other works

Compared *to tf, tf-isf*, and the term weighting scheme proposed in [2], our scheme can more accurately score and identify hot terms. Furthermore, our scheme models the short term burst and historical variation correctly. By comparing with real world data and other approaches, our system can extract hot news subjects effectively with higher recall. Furthermore, the result is relatively stable through continuous observation over one week.

VI. Conclusions

We propose herein a real-time hot news recommendation system to answer the question from users: "what is the hot news today?" In other words, given all news articles on the current day *p* and previous news archives, identify the hot news on day *p*.

We show two key characteristics of hotness: short-term burst and large historical variation. By incorporating the two important features of hotness, we also proposed a novel term weighting scheme for hotness to extract hot terms on given date. In order to compute the historical variation of a term correctly, we examined a lot of related factors and modeled the historical variation of a term by variation of interval (in day) between two occurrences of this term. With each news article on day *p* represented by hot terms it contains, the articles are grouped into clusters which represent hot events on that day.

We foresee the system to show on the internet in real time the hot topics over news feeds or other document corpora indicated, as shown graphically in the above figures. This will necessitate real time applications of the formulations we have developed to lead to the visualization shown here in, marking with different colors and icons the hot topics. The hot terms and hot new articles extracted are different from day to day, and reflect the hot events on each day.

References

[1] Seokkyung Chung, Dennis Mcleod, Dynamic topic mining from a news stream, In Proceedings of the 2nd International Conference on Ontologies, Databases, and Applications of Semantics (ODBASE 2003), Sicily, Italy.

[2] Gianna M. Del Corso, Antonio Gull′, Francesco Romani. Ranking a Stream of News, WWW 2005, May 10-14, 2005, Chiba, Japan.

[3] Kuan-Yu Chen, L. Luesukprasert, S.-c.T. Chou, Hot topic extraction based on timeline analysis and multidimensional sentence modeling, IEEE Transactions on Knowledge and data engineering, Volume 19, Issue 8, August, 2007.

[4] James Allan, Rahul Gupta, Vikas Khandelwal: Temporal Summaries of News Topics, SIGIR 2001.

[5] Jain, A.K., and Dubes, R.C.,1988, *Algorithms for Clustering Data*, Prentice Hall: New Jersey.

[6] Salton, Gerard and Buckley, C. (1988). "Term-weighting approaches in automatic text retrieval". Information Processing & Management

[7] Joel Larocca, Neto Alexandre, D. Santos, Celso A. A, Kaestner Alex, A. Freitas, Catolica Parana, Document Clustering and Text Summarization, 4th International Conference Practical Applications of Knowledge Discovery and Data Mining (PADD-2000)

[8] Khoo Khyou, Bun Mitsuru Ishizuka, Topic extraction from news archive using TF*PDF algorithm, Proc. Third Int'l Conf. Web Information Systems Eng, 2002.

[9] http://incubator.apache.org/uima/

[10] http://alias-i.com/lingpipe/

[11] Lan You, Xuanjing Huang, Lide Wu, Hao Yu, Jun Wang, Fumihito Nishino, Exploring Various Features to Optimize Hot Topic Retrieval on WEB, ISNN (1) 2004, 1025-1031

[12] Malu Castellanos, HotMiner: Discovering Hot Topics on the Web". Proceedings of SIAM workshop on Text Mining, Arlington, Virginia, April 2002.

[13] http://en.wikipedia.org/

Term	Definition
Term	Words and compound words (but not phrases) that are used in specific contexts. [13]
Entity	Atomic elements in text such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc. Each entity has its name and is referred to by a term. [22]
News	News is any new information or information on current events which is presented by print, broadcast, Internet, or word of mouth to a third party or mass audience. [22]
Topic	A broader category of events, or a class of events, such as "volcanic eruption" [4]

Event	Some unique thing that happens at some point in time, such as “the eruption of Mount Pinatubo on June 15th, 1991” [4]

Fig. 1. Term definitions

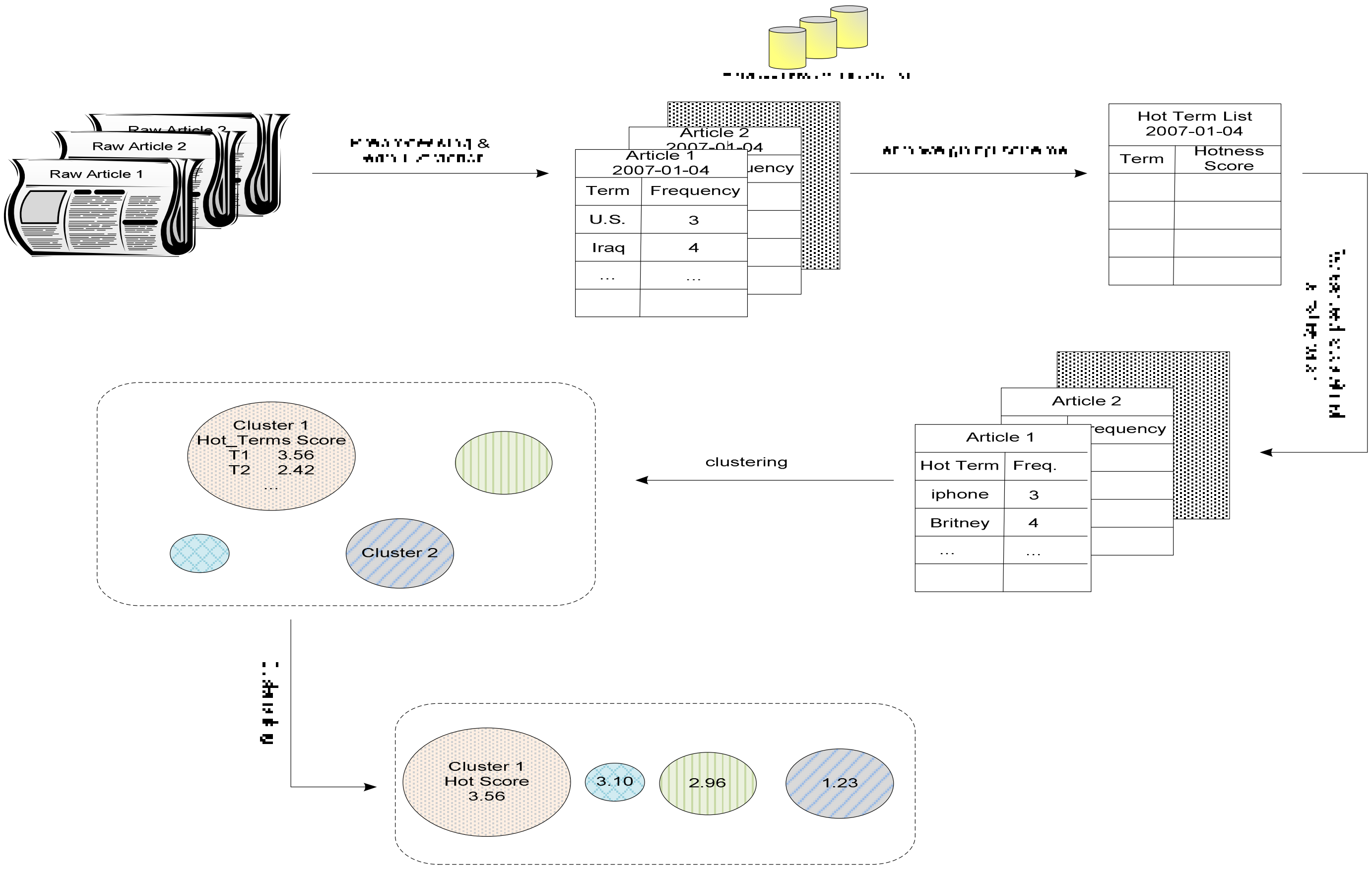

Fig. 2. Overview of our real-time hot news extraction system

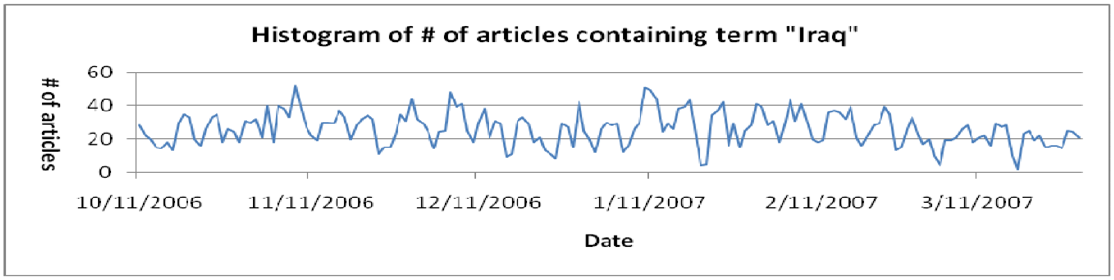

Fig. 3. Histogram of # of articles containing term “Iraq”

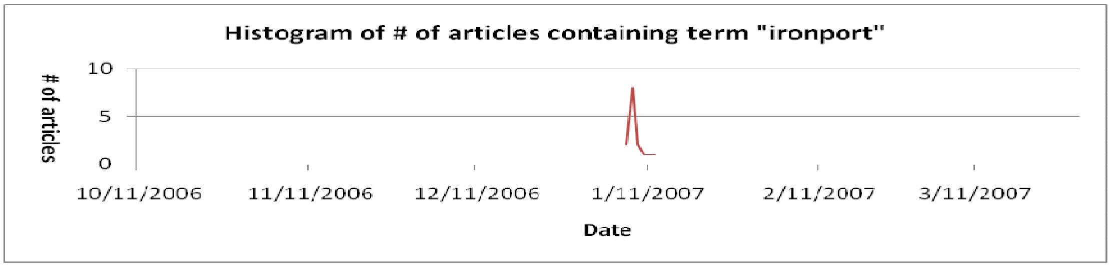

Fig. 4. Histogram of # of articles containing term "ironport"

Hot Term	Hotness Score	Frequency	Term Weight
term1	3.22	3 /(3+4+2+3)	3.22 * 3 / 12
term2	3.01	4/ (3+4+2+3)	3.01 * 4 / 12
term3	1.98	2/ (3+4+2+3)	1.98 * 2 / 12
term4	1.45	3/ (3+4+2+3)	1.45 * 3 / 12
…			

Fig. 5. Our representation of an article after hot terms are extracted

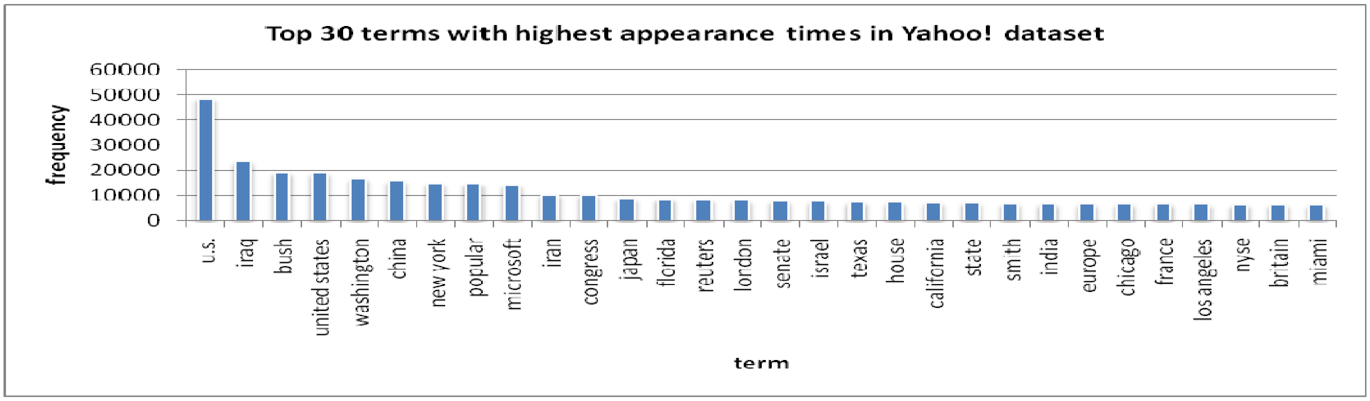

Fig. 6. Top 50 terms with highest appearance times in Yahoo! dataset

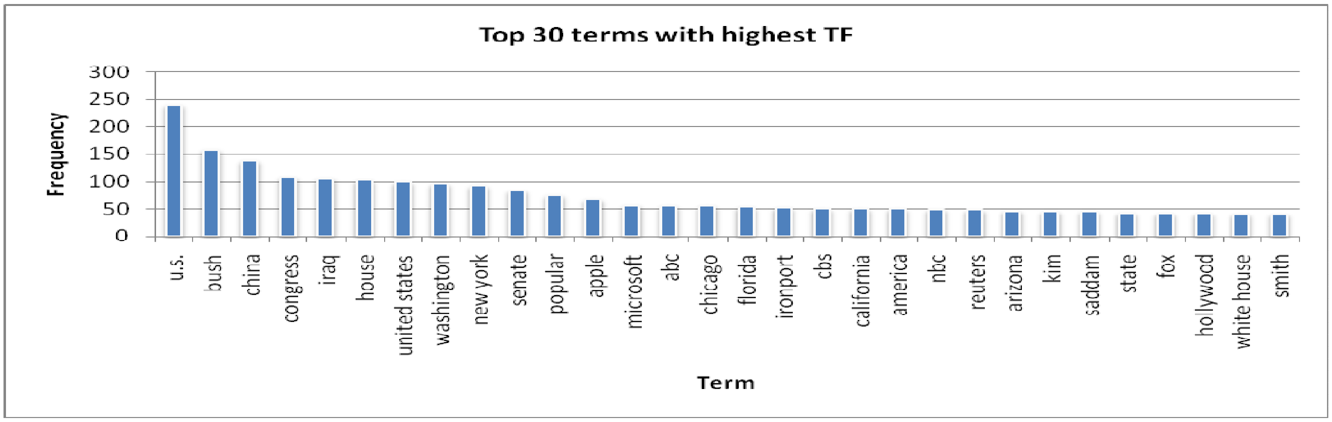

Fig. 7. Top 50 terms with highest TF

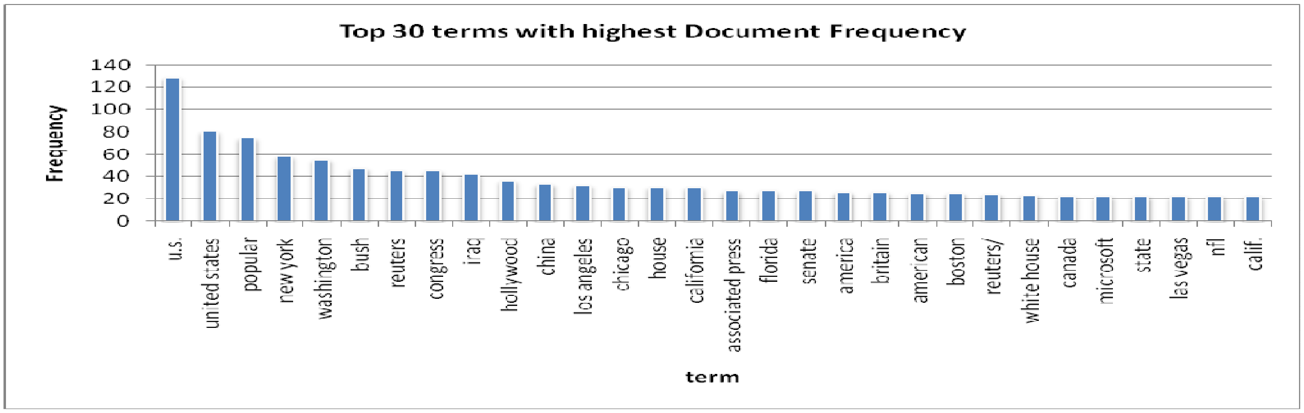

Fig. 8. Top 50 terms with highest Document Frequency

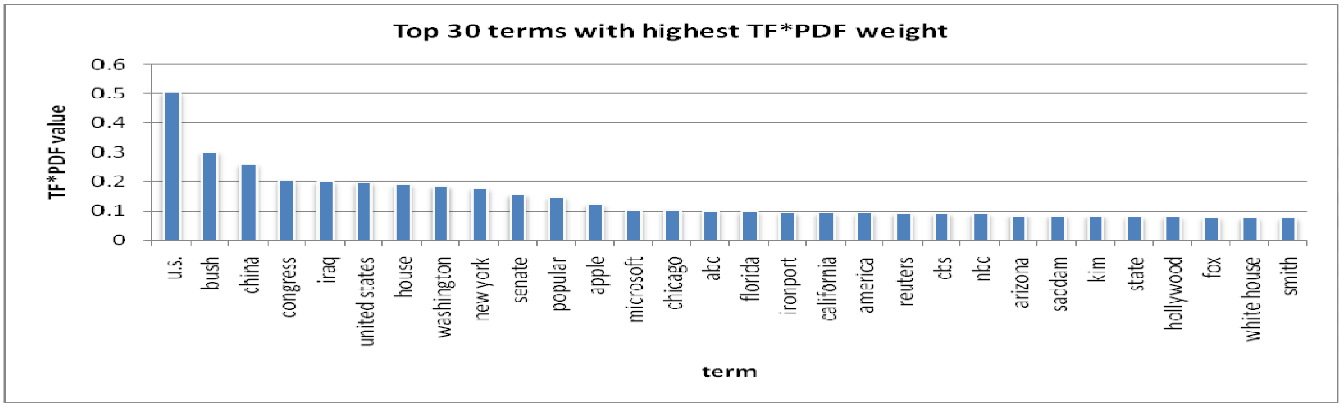

Fig. 9. Top 50 terms with highest TF*PDF weight

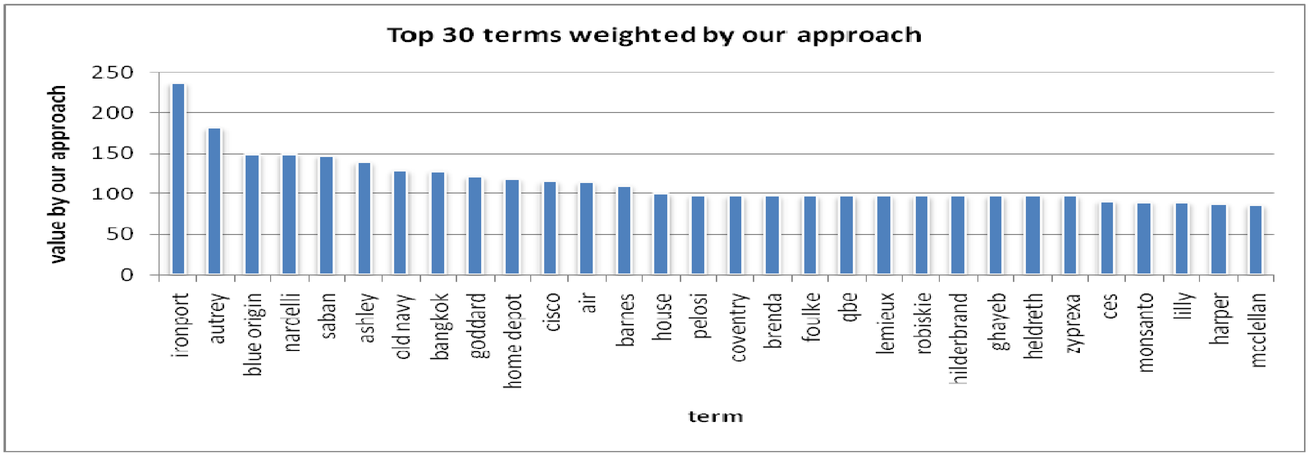

Fig. 10. Top 50 terms weighted by our approach

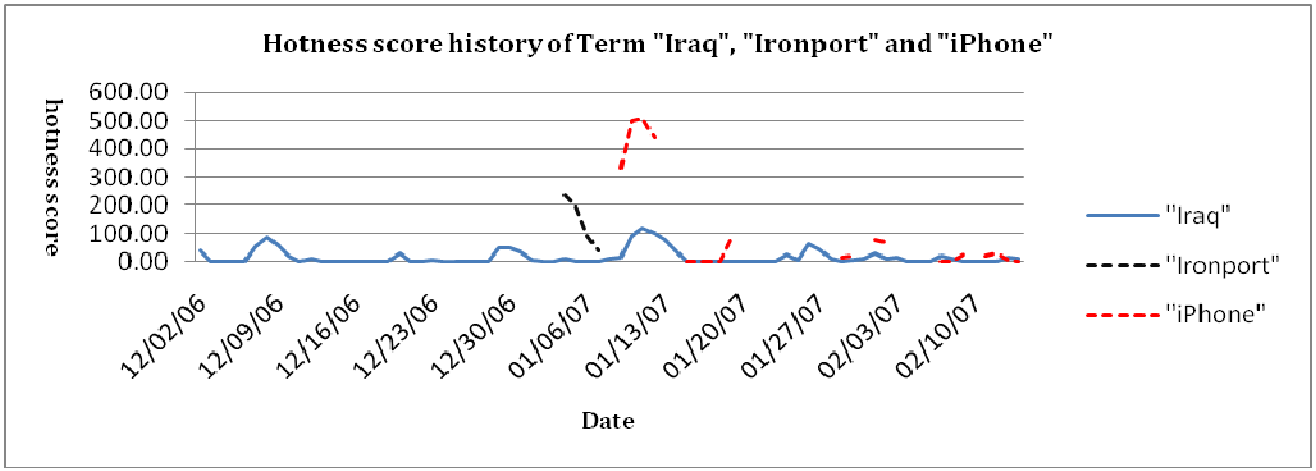

Fig. 11. Hotness score history of three hottest terms