Information Fusion 20 (2014) 129–135

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Detecting trends on the Web: A multidisciplinary approach

Rodrigo Dueñas-Fernández [a], Juan D. Velásquez [a,*], Gaston L'Huillier [b]

[a] Department of Industrial Engineering, Universidad de Chile, Av. República 701, P.O. Box: 8370439, Santiago, Chile
[b] Groupon Inc., 3101 Park Blvd., Palo Alto, CA, USA

ARTICLE INFO

Article history:
Received 6 August 2013
Received in revised form 19 November 2013
Accepted 6 January 2014
Available online 2 February 2014

Keywords:
Trend detection
Web opinion mining
Topic modeling

ABSTRACT

This paper introduces a framework for trend modeling and detection on the Web through the usage of Opinion Mining and Topic Modeling tools based on the fusion of freely available information. This framework consists of a four step model that runs periodically: crawl a set of predefined sources of documents; search for potential sources and extract topics from the retrieved documents; retrieve opinionated documents from social networks for each detected topic and extract sentiment information from them. The proposed framework was applied to a set of 20 sources of documents over a period of 8 months. After the analysis period and that the proposed experiments were run, an F-Measure of 0.56 was obtained for the detection of significant events, implying that the proposed framework is a feasible model of how trends could be represented through the analysis of documents freely available on the Web.

1. Introduction

In the current state of the art, the only widely used method for inferring trends and trying to assert if a topic is becoming a trend is to run surveys over a sparse set of individuals. Nevertheless, the use of surveys for such purpose has some caveats that need to be addressed in order for it to become a more useful tool than what it is nowadays. For example, it is highly likely that the background and attitude of the interviewers interfere with the results of the survey itself, thus making the results biased [1]. Therefore, there is an unfulfilled need to complement the existing methodologies for trend detection and any other traditional approach focused on gathering knowledge regarding events that are occurring daily. In particular, the usage of the freely available information that exists on the Web allows us to access a significant number of people freely expressing their opinion [2] that could not have been reached otherwise.

In this study we propose a generic framework that allows using the data available on the Web towards the detection and modeling of trends over time. With that objective in mind, a multidimensional approach was taken, where a topic will be monitored over time according to how much media coverage it gets and how the users of social networks react to it. Thus, the proposed framework consists of two main components, the one focused on extracting topics and monitoring their evolution in the traditional media, and the second one in charge of extracting valuable information about how users feel and the opinions they publish on social networks.

To the best of our knowledge, there are no unified methodologies that try to tackle the trend detection problem from a multidisciplinary approach, i.e. that take into consideration both the problems of Topic Modeling and Opinion Mining. Nevertheless, every problem mentioned above within the framework of trend detection has been approached over the last years by different disciplines. The branch of knowledge that involves both issues is Information Retrieval, which focuses on retrieving documents from the Web and process them to be able to analyze the data contained within them.

Under its practical and theoretical frameworks, information retrieval has presented several techniques that allow the retrieval and processing of relevant documents. In terms of detecting the topics of such documents and the specific events or features about which opinions have been expressed, text mining and natural language processing communities have developed a vast set of models to determine what are the topics being discussed across a collection of documents [3]. Finally, the field of Web Opinion Mining has presented several approaches to represent the polarity of documents posted on the Web by their users [4], whether they come from traditional media or social media, the latter usually having less structured content. The main contribution of this work is to integrate these disciplines into one unified framework that allows us to monitor trends on the Web using information present in both the traditional media and the social media (such as social networks).

* Corresponding author. Tel.: +56 2 2978 4834; fax: +56 2 2689 7895.
E-mail addresses: rduenas@ing.uchile.cl (R. Dueñas-Fernández), jvelasqu@dii.uchile.cl (J.D. Velásquez), gaston@groupon.com (G. L'Huillier).
URL: http://wi.dii.uchile.cl/ (J.D. Velásquez).

http://dx.doi.org/10.1016/j.inffus.2014.01.006
1566-2535/

This paper is organized as follows. In Section 2 a brief summary of related research is provided. Section 3 describes the proposed methodology for detecting trends on the Web. Section 4 outlines the experiments performed with the proposed methodology and Section 5 provides some conclusions and suggests future research.

2. Related work

Although there is no unique definition of what a trend is, how it should be represented and how a topic becomes a trend, several approaches have been proposed by multiple authors to detect trends or the evolution of topics over time in specific areas. For example, applications in politics are presented in [5,6] and finance in [7]. Due to the broad definition of what a trend is, there have not been a significant number of attempts to develop generic frameworks that go beyond a singular application domain, or even monitor how topics evolve in multiple areas. A basic example is presented in [8], where the main focus of their research is to show an approach towards building a trend detection framework on top of a cloud computing architecture, rather than proposing a framework capable of retrieving documents and deciding whether a topic presented in several documents over time reflects a trend or not.

In terms of information retrieval, there is a vast amount of literature that provide some insights about the different types of data and how this data should be handled [9]. Research has been done in information retrieval frameworks for the detection of trends in blogging [10], microblogging (e.g. Twitter) [11] and social networking sites (e.g. Facebook) [12]. Once the data has been retrieved and stored, the textual information has to be processed in such a way that the underlying patterns are extracted for further usage. In this domain, keyword-based analysis approaches have been proposed in the specific context of Web usage mining [13,14]. Also, several approaches focused on the detection of unknown events or determining the impact of news online [15,16] have been proposed.

However, one of most relevant techniques used in recent years for modeling how events evolve over time are topic models [17]. A topic model can be considered as a probabilistic model that relates documents and words through variables, which represent these main topics, inferred from the text itself. In this context, a document can be considered as a mixture of topics, represented by probability distributions that generate the words that belong to a document that contains these topics. The process of inferring the latent variables, or topics, is the key component of this model, whose main objective is to learn the distribution of the underlying topics from text in a given corpus of text documents. A main topic model is the latent Dirichlet allocation (LDA) [18]. LDA is a Bayesian model in which latent topics of documents are inferred from estimated probability distributions over a training data set.

Opinion mining and sentiment analysis is a field whose objective is to consider a collection of opinionated documents and determine the orientation (positive, negative, and objective) of an opinion about a particular aspect of an entity at a given time [19]. In terms of trends detection, this task is fundamental in order to identify whether the trending topics are being generated with a certain opinion orientation. In this work, the opinion mining step will be focused in the usage of lexicon-based algorithms. Algorithms that are based on the use of lexicons can be found in [1,2], which according to the research presented in [20] can return valuable information in the context of mining opinions of documents retrieved from microblogging sites. It should be noted that opinion mining algorithms that focus on the classification of polarity face several challenges, such as irony and sarcasm linguistics [21,22] and also the amount of text available in a document [23].

3. A methodology for trend detection on the Web

In this chapter, the methodology proposed to detect trends on the Web is presented. First, the definition of the problem to solve and every term used throughout this paper are detailed. Next, some of the main text analysis techniques used during the development of the proposed methodology for detection of trends on the Web are discussed. Finally, the methodology itself and the main contribution of this work are described.

3.1. Problem definition and general notation

In the following, the term *Trend* will be presented together with its ontological and linguistic representation. In this context, a *trend* will be defined as a given event whose impact on a system as a whole, is above the average over a certain period of time.

The problem of detecting trends on the Web is described. Several research areas are focused on modeling the so-called collective behavior in order to, for example, predict how important events will develop, which politician will win a debate, which football team will win a match and so on.

Even though existing methodologies to predict trends and monitor their evolution over time have been successfully applied to a large variety of problems, there's a vast amount of information not being used, created by users on the Internet, where the act of expressing one's opinion or feelings is not restrained by the common issues that are found in the standard methodologies such as limited time and biased answers based on the person running the interview.

The objective of this research is to tackle what will be called as the *trend detection problem*, which is defined as:

Definition 1 (*Trend Detection Problem*). Given a set of topics, to determine if the way they behave over time makes them qualify as a trend.

In this research the following definition of *trend* will be used:

Definition 2 (*Trend*). A trend is a given event or topic whose impact on a system as a whole, is above the average over a certain period of time. Furthermore, for a system composed by a chain of events, a trend is defined by its expected future behavior given how it behaved in the past and how it reacts to external stimulus.

In this study, a *factual document* is a document that contains no opinion whatsoever and refers to one or more events. On the other hand, the term *opinionated document* refers to any opinionated document whose subject is an event. Examples of these types of documents are tweets and opinion columns in journals, among other personal opinions expressed on the Web by one of its users.

As the detection of trends in most useful scenarios is always framed within a certain domain of knowledge, a similar approach will be taken in our methodology, where the set of websites to be crawled for documents is defined beforehand and they are expected to belong to specific domain of knowledge. Each of these sources of documents will be referred to as *feeds*.

In order to be able to create a more descriptive model of topic evolution, an information fusion [24] approach was taken, in which their evolution is measured by a multidimensional analysis based on information retrieved from *factual documents* and *opinionated documents* extracted from several sources. The proposed methodology for detecting and modeling trends on the Web consists of four main steps that are executed periodically and then complemented by the visualization of the extracted data. These steps are:

1. Crawl every feed and extract every factual document found.
2. Using a topic model, infer the underlying topic structure for the factual documents retrieved during the previous step and link them with the ones extracted in past periods.
3. Evaluate if there is any potential feed that could be included in the current set of crawled feeds.
4. Retrieve opinionated documents and extract sentiment information for every topic being discussed on the current period.

3.2. Detecting topics towards an opinion mining analysis

Based on the definition given of the trend detection problem, to detect trends on the Web the first step that needs to be accomplished is to extract the topics that are being discussed within the subset of the web containing factual documents. In order to do so, and be able to gather the needed data from social network websites to perform a sentiment analysis, a crawling algorithm is used on the documents retrieved from such websites.

Algorithm 1. Document Retrieval

Input: $\{d_i\}_{i=1\ldots\mathbf{N}}$
Output: $\{\vec{d}_i\}_{i\in\mathbf{N}}$
1: documents := []
2: **for all** $f \in \{F_i\}_{i\in\mathbb{N}}$ **do**
3: document ← `retrieveDocument`(f)
4: documents ← documents $\bigcup$ document
5: **end for**
6: **return** documents

Given a set of feeds, a simple crawling algorithm shown in Algorithm 1 was used to retrieve the raw documents from each feed. Documents retrieved by this crawling algorithm are stored as raw data together with all the metadata that could be extracted from the feed that it came from. Some of the information present as metadata in these feeds are categories and labels used on the site to classify content, author, language and original publication date.

Once the documents are retrieved from each feed, an LDA [18] model is used to extract the underlying structures for the topics that are present on them. This model allows, given a collection of documents $\{d_i\}_{i=1\ldots\mathbf{N}}$, obtain a set of topics $\{t_i\}_{i=1\in\mathbf{N}}$ described by the probability $P(topic = t|document = d)$ for a document d to discuss topic t and, for each pair of words and topics (w, t), the probability $P(topic = t|word = w)$ for a word w to describe a topic t.

To achieve a representation of how topics evolve over time is necessary to extract a set of topics for each period t_i and link these with the topics of the previous period and so on. One of the limitations of the LDA model is that it does not correlate topics over time; therefore, it is mandatory to create a way to correlate topics extracted during a period t with the topics extracted from documents retrieved in past periods. The approach proposed for this research is the following:

1. For every period t, collect the documents from the two preceding periods t_{i-1}, t_{i-2} and use them as training data for a new LDA model.
2. Then, using the trained model a Bayesian inference is performed over the set of documents retrieved in period t. This is done in order to discover its underlying topic structure.

Once every document published in periods t_i, t_{i-1}, t_{i-2} is retrieved and the topic structure that represents the documents retrieved in t_i is inferred, it is possible to link two topics T y $T\prime$, with corresponding word-topic probability vectors $\vec{w}_T$ and $\vec{w}_{T\prime}$ making use of a distance function defined as shown in Eq. (1):

$$d(T, T') = \sum_{w_i\in\vec{w}_T}\sum_{w_j\in\vec{w}_{T'}} w_i - w_j \tag{1}$$

Then, for each pair T, T' of topics, a link is created if and only if the result of the function $d(T, T')$ is below a threshold ϕ defined at the beginning of the analysis.

3.3. Extracting sentiment information focused on trends detection

Once a period is over, it is necessary to complement the factual information extracted on the previous step with sentiment information extracted from opinionated documents retrieved from social networks. Even though there are many sources for opinionated documents, the ones that reflect more clearly if a topic is trending or not are those present in social networks.

In order to extract opinions from such documents, an algorithm based on lexicon data is used. The usage of lexicons in opinion mining models is based on the hypothesis that a word can be considered as a fundamental knowledge unit of an opinion, and therefore it can shed some light on the sentiment polarity of a document as a whole.

In this research, the *SentiWordNet* [25] platform is used as a resource of lexical information, in which the labeled information is described as:

$$\vec{w} = \langle w, w^p, w^o\rangle \tag{2}$$

With $\vec{w}$ the labeled vector for the word w, w^p its positive sentiment score, w^n its negative sentiment score and w^o its objectivity score. Furthermore, every labeled word in *SentiWordNet* fulfills Eq. (3):

$$w^p + w^n + w^o = 1 \tag{3}$$

Thus, given a set $\vec{w}_d$ of size k consisting of every word present in an opinionated document d, it is possible to associate their sentiment scores with the document as shown in Eq. (4):

$$d^p = \frac{\sum_{i=1}^{k} w^p}{\|\vec{w}_d\|}, \quad d^n = \frac{\sum_{i=1}^{k} w^n}{\|\vec{w}_d\|}, \quad d^o = \frac{\sum_{i=1}^{k} w^o}{\|\vec{w}_d\|} \tag{4}$$

Then, considering a method `polarity (document)` that given an opinionated document d returns its sentiment vector (d^p, d^n, d^o) and a set $\mathcal{D}_\tau = \{d_i\}_{i=1\ldots\mathbf{N}}$ of opinionated documents related to a topic τ; the sentiment score for a topic τ given the set of opinionated documents $\mathcal{D}_\tau$ can be calculated by using Eq. (5):

$$\vec{o}_\tau = (o^p_\tau, o^n_\tau) = \left(\frac{\sum o^p_d}{\|\mathcal{D}_t\|}, \frac{\sum o^n_d}{\|\mathcal{D}_t\|}\right) \tag{5}$$

To determine which documents will be retrieved from the social networks being mined, a simple permutation is used to generate the queries. In this case, for a given topic τ, the queries correspond to all the *n-grams* of length n that can be formed by the keywords which describe it during the period t. In particular, our research will focus solely on Twitter as the social network to be mined.

3.4. Expanding the set of crawled feeds

Most of the retrieved documents on the crawling phase contain hyperlinks pointing to different websites that talk about the same topics that are being discussed on them. Therefore, this new set of information allows the inclusion of new elements to the set of crawlable feeds.

Several approaches have been developed to allow the discovering of blog communities based on the relevance of the content

published among a given set of blogs [26,27]. Given that the presented methodology focuses on detecting trends in a given domain of knowledge, it is expected that blogs discussing topics belonging to any given domain can be grouped in a blog community.

As such, we propose a methodology for expanding the set of feeds being mined that consists of two steps: the first step is shown in Algorithm 2 which detects a set of potentially useful feeds based on how frequently they are mentioned in the documents already retrieved; and the second step which focuses on evaluating each potential feed to see if they belong to a similar blog community in order to determine if their contents could add valuable information to the topic mining algorithm.

A *potential feed* is defined as a feed that contains information related to the topics being discussed on the previously defined set of feeds. These feeds are evaluated later in order to decide if they should be included in the set of feeds being crawled.

The method `extractFeedURLs` extracts all URLs of a document. As these documents are published in blogs that are financed by advertising, many of these URLs correspond to ads and they should be ignored as they will never provide useful information. Furthermore, taking the complete URL, or just taking the domain is not enough as our objective is to detect potential additions to our feed set. In order to avoid these issues, a set of URL stemming rules is defined:

- If the URL has a *query* component, it must be removed. The *query* component of a URL is the one that comes after a question mark `?` and contains information to be sent to the server, such as marketing campaign information, and search queries.
- If the URL points directly to a file (e.g. `html`, `pdf`, `php`) only the domain name will be used.

Algorithm 2. Detection of potential sources

Input: $\{d_i\}_{i\in\mathbb{N}}$
feasibleFeeds = []
for all document $\in \{d_i\}_{i\in\mathbb{N}}$ **do**
  feeds = extractFeedURLs (document)
  for all feed $\in$ feeds **do**
    if database.updateFeedCount (feed) **then**
      feasibleFeeds.append (feed)
    end if
  end for
end for
average = database.getFeedCountAverage (feasibleFeeds)
for all feed $\in$ database.getFeedData (feasibleFeeds) **do**
  if feed.count $>$ average **then**
    createPossibleNewFeed (feed)
  end if
end for

Only the number of feeds that show a given URL is considered in our proposed approach as it outperformed a frequency-based approach. The reason behind this is that given the different styles of citation and hyperlinking used by different blogs, if a frequency-based approach was considered, the potential feed selection algorithm became biased towards the URLs shown in those feeds that had a more aggressive citation style (i.e. they added a lot of hyperlinks to a document) than in those with a more passive citation style (i.e. those who add a couple of citations at the end of the document).

Function `database.updateFeedCount (feed)` increases by one as the source appears in the data, and returns `true` if the source has not been yet moved to the list of sources to be evaluated, or `false` otherwise.

Function `getFeedCountAverage (feeds)` is in charge of getting the average between all the successful appearances of the input URLs. Then, function `createPossibleNewFeed (feed)` creates a new entry in the list of candidate sources to be evaluated on the second step of this methodology, and marks the new source as processed so it will be ignored in future iterations. This way, it will only consider the list of potential sources to be all the URLs that have a frequency higher than the average.

To evaluate these feasible feeds, a variation of the weblog communities discovery algorithm by Bulters et al. [26] focused on using topic information to create communities will be used.

Once a feed has been added to the feasible feed set, the algorithm starts to crawl it but the stored documents will not be used by any of the previously mentioned phases. Then, its *linkStrength* (step 2 of the methodology in [26]) shown in Eq. (6) is calculated between the feasible feed f and each one of the feeds f' being used to extract topic information. If the number of feeds that have a *linkStrength* greater than σ (using as an origin point the candidate feed) is greater than ρ, the feed will be added to the set of processed feeds.

$$linkStrength(f,f') = w_{relev}\cdot relev + w_{reciprocity}\cdot recip + w_{cocitation}\cdot cocit \tag{6}$$

The relevance, reciprocity, and cocitation terms are defined by Eqs. (7)–(9) respectively. A document d contains relevant content if it contains a certain percentage of the top N keywords of a topic t, for any element of the set of topics $\{t_i\}_{i\in\mathbb{N}}$ that belong to the documents retrieved from $f\prime$. Also, let r_d be 1 if a document d is relevant, 0 otherwise.

$$relev = \frac{\sum_{d\in D_f} r_d}{\|D\|} \tag{7}$$

$$recip = \begin{cases} 1.0 & \text{if } f'.linkSet \text{ has a link to } f \\ 0.0 & \text{otherwise} \end{cases} \tag{8}$$

$$cocit = \frac{\|f.linkSet \cap f'.linkSet\|}{\|f.linkSet\|} \tag{9}$$

The weights used for calculating the *linkStrength* Eq. (6) are those approximated in [26], which are $w_{relev} = 0.5, w_{co-citation} = 0.3$, and $w_{reciprocity} = 0.2$.

This methodology is described in Algorithm 3, which receives as input parameters the potential source F_p to evaluate and the threshold value ρ that will be used to decide whether F_p will be included into the set of analyzed sources.

Algorithm 3. Evaluation of potential sources

Input: F_p, ρ
relatedFeeds = 0
actualFeeds = database.getFeeds ()
for all *feed* $\in$ *actualFeeds* **do**
  if $linkStrength(F_p, feed) > \sigma$ **then**
    relatedFeeds++
  end if
  if $\left(\frac{relatedFeeds}{actualFeeds.length}\right) > \rho$ **then**
    database.addFeed (F_p)
  end if
end for

4. Experimental results

The described methodology was applied to a set of 20 feeds discussing technology topics over a period of eight months. *RSS* (Real Simple Syndication) feeds were used because they show the most complete amount of metadata, and also because the way documents are presented in an *RSS* feed is easy to process and allows passive polling for new documents without abusing the servers of our content providers.

For each retrieved document, the following information was stored: original content in *HTML* format, published date, original URL, publishing feed, creation timestamp and any metadata contained within the *RSS* entry.

4.1. Structure and content processing

Every document retrieved by the crawling processes is stored as raw data (i.e. with *HTML* tags, external links, navigation links, etc.) and prior to being used by both the topic modeling and opinion mining algorithms they are pre-processed through standard data cleaning methodologies such as the removal of *HTML* elements, extraction of *stop-words* and stemming. The crawler used for retrieving factual documents possesses the capability of updating documents if they change after they were initially stored, if these changes were explicitly registered by the feed being mined, in order to obtain a more realistic representation of the source.

4.2. Feed set expansion algorithm

The objective of this experiment is to measure the effectiveness of the relevance classification algorithm for feasible feeds. Thus, it is necessary to determine the existence of a relationship between the feed being evaluated and the initial set of feeds. To assert if a relationship exists between them, a manual analysis of the topics discussed in each feed was performed.

4.2.1. Discovery of feasible feeds

To evaluate the algorithm for discovering feasible feeds, every feed present in a set of retrieved documents was manually classified as relevant or not relevant. The criteria used to define a feasible feed as relevant was if the content published by the feed pertains to the same area of knowledge as the feeds being mined. For this experiment the criteria used was if these documents discuss any kind of technology-related events or entities.

To evaluate the algorithm that creates the set of feasible feeds, the following terms are defined:

1. *RPSS* = Relevant potential sources selected to be evaluated.
2. *PSE* = Potential sources to be evaluated.
3. *RPS* = Relevant potential sources.

Then the precision and recall that evaluates the quality of Algorithm 2,

$$Precision_{\text{Sources}} = \frac{RPSS}{PSE} \tag{10}$$

$$Recall_{\text{Sources}} = \frac{RPSS}{RPS} \tag{11}$$

4.2.2. Evaluation of feasible feeds

Once the set of feasible feeds was determined, and they were crawled during a period of two weeks, the algorithm for evaluating potential feeds was run and then they were manually classified as relevant or not relevant.

Finally, each potential feed was crawled during a period of two weeks and the relevance classification algorithm was applied to each one of them using as input data the documents retrieved during this period.

Let,

1. *RSUAC* = Relevant sources under analysis classified as relevant.
2. *SUA* = Sources under analysis classified as relevant.
3. *RSUA* = Relevant sources under analysis.

To evaluate the relevance classification algorithm the metrics shown in (12) and (13) were used.

$$Precision_{\text{Analysis}} = \frac{RSUAC}{SUA} \tag{12}$$

$$Recall_{\text{Analysis}} = \frac{RSUAC}{RSUA} \tag{13}$$

4.2.3. Experiment results

The results of the experiment for the discovery of feasible feeds are shown in Tables 1 and 2. A total of 12,000 documents were chosen and 31,778 relevant links were extracted, from which 1493 correspond to unique feeds. These links were distributed as follows:

The average of document-citations for these links is 2.4, thus 180 feeds are included in the set of feasible feeds. Of these feasible feeds, 79 correspond to websites of services, products or brands, and 101 to blogs, news sites or similar websites where 61 of them published mainly technology-related articles, and the rest of them published general interest news that included technology articles.

The $Recall_{\text{source}}$ of Algorithm 3, using Eq. (11), is 0.35. Even though its recall is low because the number of feasible feeds tends to increase as the evaluation period increases, as the majority of these feeds only appear in one or two documents, it can be considered that this low recall does not imply a loss of valuable information. In fact, if the $Recall_{\text{source}}$ is calculated without considering those feeds that show up in only one document, it goes up to 0.58. Furthermore, the $Precision_{\text{source}}$ using Eq. (10) is 0.56 mainly due to the quantity of websites of services and products which many blogs in the technology area use to mention either a product launch or a review.

The experiment proposed to evaluate the relevance classification algorithm was run with multiple values of ρ. Its precision (Eq. 12) and recall (Eq. 13) are shown in Table 3

In Table 3 it can be observed that neither the recall nor the precision of the algorithm could be calculated if a high enough value of ρ was used because no feed was classified as relevant. Furthermore, the *precision* of this algorithm increases with higher values of ρ due to the higher requirements for the feed to be more related with technology, and on the other hand, the *recall* decreases because the number of selected feeds is lower due to the higher restrictions. To pick an optimal value for ρ, the one with

Table 1
Distribution of links by type.

Type	Quantity
Feeds being mined	27,740
File	263
Social Networks (Facebook, Twitter, etc.)	397
Streaming Sites	45
Government Websites	122
Encyclopedic Websites (Wikipedia, IMDB, etc.)	234
Feed Aggregators	192
University Websites	33
Others	2752

Table 2
Number of potential feeds grouped by amount of documents mentioning them.

Document citations	Potential feeds
1	1061
2	232
3–10	143
11–100	32
100 or more	5
Feeds being mined	20

Table 3
Precision and *Recall* for multiple values of ρ.

ρ	Precision	Recall
0.3	0.32	0.51
0.5	0.53	0.42
0.6	0.57	0.30
0.8	–	–

the best precision possible must be used, because if a wrong feed is included, it has a high chance of introducing noise or incorrect data in the models causing a decreasing performance over time. Even if picking the best precision possible implies a lower recall as seen in Table 3, as long as relevant feeds are being included the algorithm is useful.

4.3. Model validation and trend visualization

The purpose of evaluating this model is to determine its capability of representing how media react towards events that occur during the period when the analysis is being done. The events focused on by this research will be called as *significant event* and are defined by:

Definition 3 (*Significant Event*). If between two consecutive periods t_i and t_{i+1}, the difference between the number of factual documents published $\frac{\|\vec{D}_{t_{i+1}}\| - \|\vec{D}_{t_i}\|}{\|\vec{D}_{t_i}\|}$ is greater than a threshold ρ, then a significant event occurred in t_{i+1}.

An example of a topic containing a significant event can be seen in Fig. 1, in which the sentiment associated with it is shown as a spline, and the number of factual documents where the topic is mentioned. Given the big increase in factual documents between periods 5 and 6, a significant event is marked in period 6. Furthermore, it can be seen how the media coverage and the sentiment on social networks change over time.

To evaluate the proposed framework, the following approach was taken: for each topic, their corresponding time series will be manually analyzed for *significant events*, and the precision of the framework will be the precision of the algorithm regarding the number of significant events, i.e. if a major event happened in the same period as a *significant event* that is shown by the methodology, then it is counted as a success. Let,

1. *SECC* = Significant events correctly classified as such (manual annotation).
2. *ASEM* = Number of significant events found by the algorithm.
3. *ASEF* = Number of significant events found manually.

Events that could be considered as "significant events" are manually annotated. The precision of the methodology is calculated by the Eq. (14), recall by Eq. (15), and F-measure by Eq. (16).

$$Precision_{SE} = \frac{SECC}{ASEM} \tag{14}$$

$$Recall_{SE} = \frac{SECC}{ASEF} \tag{15}$$

$$FMeasure_{SE} = 2 \cdot \frac{Precision_{SE} \cdot Recall_{SE}}{Precision_{SE} + Recall_{SE}} \tag{16}$$

4.3.1. Experimental results

The proposed methodology was executed over a period of eight months, during which a total of 200,890 factual documents were collected, out of which 117 topics were extracted, and 268,800 tweets were retrieved. Also, a total of 65 *significant events* distributed over these topics were manually detected. To avoid getting incorrect results, only significant events that were detected after six documents were retrieved in a specific period of a topic were used for these calculations.

As shown in Table 4, for values of ρ of 0.6 or higher no significant events were found. This is because that given the amount of news published on a weekly basis by feeds which discuss technology topics, the threshold of new documents needed to qualify as a significant event cannot be met.

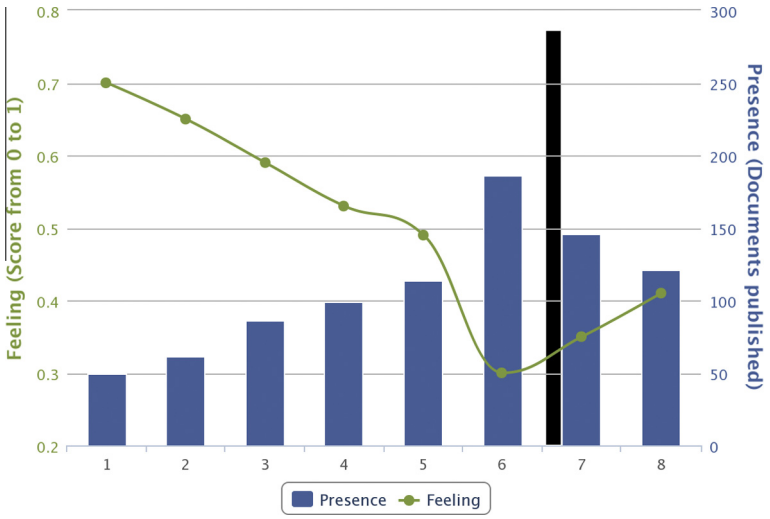

Fig. 1. Evolution of a topic over time.

Table 4
$Precision_{SE}$, $Recall_{SE}$, and $FMeasure_{SE}$ for multiple values of ρ in Algorithm 3.

ρ	$Precision_{SE}$	$Recall_{SE}$	$FMeasure_{SE}$
0.2	0.25	0.71	0.37
0.3	0.38	0.65	0.48
0.4	0.48	0.57	0.52
0.5	0.61	0.51	0.56

5. Conclusion

We conclude that the methodology presented in this paper is a feasible approach to model how trends could be represented on the Web as an interaction of events, topics, and the opinions expressed by their users on social networks.

This approach takes advantage of both factual and opinionated documents on the Web to create a visual representation of topics. It allows the development of more advanced methodologies and frameworks focused on detection and modeling of trends on the Web through the extension of each component. For example, the inclusion of comments on news sites can lead to correlate how news describe a given event and the opinions expressed on the Web about it.

Given the broad definition of what a trend is and the even broader spectrum of variables that could be taken into consideration to detect them, it must be noted that this research proposes a basic approach towards this end. As such, this work is intended to be extensible and used as a framework from which several techniques could be developed.

As future research directions we propose the inclusion of an algorithm with feature detection in the opinion mining phase. Also, improving the detection of significant events will allow the platform to better detect the appearance of trends over time. Furthermore, developing metrics of correlation between the information extracted from social media and news sources would be useful. In addition, the way of recovering factual or opinionated documents could be modified towards analyzing streams of data, allowing the development of a system capable of determining in advance if significant events are going to happen, and if trends are being born.

Acknowledgements

This work was partially supported by FONDEF project D10I-1198, entitled WHALE: *Web Hypermedia Analysis Latent Environment* and the Millennium Institute on Complex Engineering Systems (ICM: P-05-004-F, CONICYT: FBO16).

References

[1] B. Ohana, B. Tierney, Sentiment classification of reviews using sentiwordnet, in: Proceedings of the 9th IT&T Conference, Dublin, Ireland, 2009, pp. 1–9.
[2] S. Brody, N. Diakopoulos, http://www.aclweb.org/anthology/D11-1052Coooooooooooooollllllllllllll!!!!!!!!!!!!!! using word lengthening to detect sentiment in microblogs, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Edinburgh, Scotland, UK, 2011, pp. 562–570 <http://www.aclweb.org/anthology/D11-1052>.
[3] A. Kao, S. Poteet, Text mining and natural language processing: Introduction for the special issue, SIGKDD Explor. Newsl. 7 (1) (2005) 1–2, http://dx.doi.org/10.1145/1089815.1089816. http://dx.doi.org/10.1145/1089815.1089816.
[4] E.M. Taylor, C. Rodriguez, J.D Velásquez, G. Ghosh, S. Banerjee, Web opinion mining and sentiment analysis, in: J.D. Velásquez, V. Palade, L.C. Jain (Eds.), Advanced Techniques in Web Intelligence, vol. 2, Springer, 2012, pp. 105–126.
[5] B. O'Connor, R. Balasubramanyan, B. Routledge, N. Smith, From tweets to polls: linking text sentiment to public opinion time series, in: Proceedings of the International AAAI Conference on Weblogs and Social Media, ICWSM '10, AAAI Press, 2010, pp. 122–129.
[6] L. Sarmento, P. Carvalho, M. Silva, E. de Oliveira, Automatic creation of a reference corpus for political opinion mining in user-generated content, in: Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion, CIKM '09, ACM, Hong Kong, China, 2009, pp. 29–36.
[7] V. Sehgal, C. Song, Sops: stock prediction using web sentiment, in: Seventh IEEE International Conference on Data Mining Workshops, 2007, ICDM Workshops 2007, ICDMW '07, IEEE Computer Society, Omaha, Nebraska, USA, 2007, pp. 21–26.
[8] A. Vakali, M. Giatsoglou, S. Antaris, Social networking trends and dynamics detection via a cloud-based framework design, in: Proceedings of the 21st International Conference Companion on World Wide Web WWW '12 Companion, ACM, New York, NY, USA, 2012, pp. 1213–1220.
[9] R.A. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
[10] N.S. Glance, M. Hurst, T. Tomokiyo, Blogpulse: automated trend discovery for weblogs, in: WWW'04: Workshop on the Weblogging Ecosystem: Aggregation, Analysis, and Dynamics, ACM, 2004.
[11] M. Mathioudakis, N. Koudas, Twittermonitor: trend detection over the twitter stream, in: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, SIGMOD '10, ACM, Indianapolis, IN, USA, 2010, pp. 1155–1158.
[12] I.P. Cvijikj, F. Michahelles, Monitoring trends on facebook, in: Proceedings of the 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, DASC '11, IEEE Computer Society, Sydney, Australia, 2011, pp. 895–902.
[13] K. Hammouda, M. Kamel, Distributed collaborative web document clustering using cluster keyphrase summaries, Inform. Fus. 9 (4) (2008) 465–480. special Issue on Web Information Fusion, doi:http://dx.doi.org/10.1016/j.inffus.2006.12.001 <http://www.sciencedirect.com/science/article/pii/S1566253506001151>.
[14] J.D. Velásquez, Web site keywords: a methodology for improving gradually the web site text content, Intell. Data Anal. 16 (2) (2012) 327–348.
[15] W. Wei, N. Cao, J.A. Gulla, H. Qu, Impactwheel: visual analysis of the impact of online news, in: Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology – Volume 01, WI-IAT '11, IEEE Computer Society, Washington, DC, USA, 2011, pp. 465–474. doi:http://dx.doi.org/10.1109/WI-IAT.2011.108 http://dx.doi.org/10.1109/WI-IAT.2011.108.
[16] J. Leskovec, L. Backstrom, J. Kleinberg, Meme-tracking and the dynamics of the news cycle, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, ACM, New York, NY, USA, 2009, pp. 497–506. doi:http://dx.doi.org/10.1145/1557019.1557077, URL http://dx.doi.org/10.1145/1557019.1557077.
[17] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning, The MIT Press, 2009.
[18] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022. <http://dl.acm.org/citation.cfm?id=944919.944937>.
[19] M. Hu, B. Liu, Opinion extraction and summarization on the web, in: Proceedings of the 21st National Conference on Artificial Intelligence, AAAI'06, vol. 2, AAAI Press, 2006, pp. 1621–1624. <http://dl.acm.org/citation.cfm?id=1597348.1597456>.
[20] E. Kouloumpis, T. Wilson, J. Moore, Twitter sentiment analysis: the good the bad and the OMG!, Artif Intell. 70 (2) (2011) 538–541. <http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/viewPDFInterstitial/2857/3251>.
[21] A. Reyes, P. Rosso, Making objective decisions from subjective data: detecting irony in customer reviews, Decis. Supp. Syst. 53 (4) (2012) 754–760, http://dx.doi.org/10.1016/j.dss.2012.05.027. http://dx.doi.org/10.1016/j.dss.2012.05.027.
[22] R. González-Ibáñez, S. Muresan, N. Wacholder, Identifying sarcasm in twitter: a closer look, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers, HLT '11, vol. 2, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 581–586 <http://dl.acm.org/citation.cfm?id=2002736.2002850>.
[23] A. Reyes, P. Rosso, T. Veale, A multidimensional approach for detecting irony in twitter, Lang. Resour. Eval. 47 (1) (2013) 239–268, http://dx.doi.org/10.1007/s10579-012-9196-x. http://dx.doi.org/10.1007/s10579-012-9196-x.
[24] J. Yao, V.V. Raghavan, Z. Wu, Web information fusion: a review of the state of the art, Inform. Fus. 9 (4) (2008) 446–449. doi:http://dx.doi.org/10.1016/j.inffus.2008.05.002. Special Issue on Web Information Fusion <http://www.sciencedirect.com/science/article/pii/S1566253508000316>.
[25] A. Esuli, F. Sebastiani, SentiWordNet: A publicly available lexical resource for opinion mining, in: Proceedings of the Third International Conference on Language Resources and Evaluation, LREC '06, European Language Resources Association (ELRA), Genoa, Italy, 2006, pp. 417–422.
[26] J. Bulters, M. de Rijke, Discovering weblog communities: a content- and topology-based approach, in: Proceedings of the International Conference on Weblogs and Social Media, ICWSM 07', AAAI, Boulder, Colorado, USA, 2007, pp. 211–214.
[27] Y. Lin, H. Sundaram, Y. Chi, J. Tatemura, B. Tseng, Discovery of blog communities based on mutual awareness, in: 15th World Wide Web Conference on Proceedings of the 3rd Annual Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 2006, pp. 1–12.