UNIVERSITY
OF TRENTO

DIPARTIMENTO DI INGEGNERIA E SCIENZA DELL'INFORMAZIONE

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.disi.unitn.it

SCALABLE DISCOVERY OF CONTRADICTING OPINIONS IN WEBLOGS

Mikalai Tsytsarau, Themis Palpanas, Kerstin Denecke, and
Marko Brosowski

July 2009

Technical Report # DISI-09-038

Scalable Discovery of Contradicting Opinions in Weblogs

Mikalai Tsytsarau *1, Themis Palpanas *2, Kerstin Denecke #3, Marko Brosowski #4

University of Trento
[1]tsytsarau@disi.unitn.eu
[2]themis@disi.unitn.eu

* *Research Center L3S*
[3]denecke@l3s.de
[4]brosowski@L3S.de

***Abstract*— Weblogs are a popular means of information communication, where people discuss a variety of topics, and often times also express their opinions on these topics. In this work, we address the problem of analyzing the evolution of community opinions across time, as these are represented in the weblogs. In particular, we are interested in identifying topics and time windows, for which contradictory opinions have been expressed.**

We describe an approach for solving the above problem, which consists of the following steps. We first introduce a technique for topic and opinion extraction that operates at the sentence level. Then, we propose a novel measure for contradictions that can effectively aggregate the relevant information from the weblog posts. We discuss its properties, and show how it can be used to detect two different types of contradictions, namely, simultaneous contradictions, and change of sentiment. Finally, we describe an efficient data structure for answering queries related to contradiction detection, and show that it has the additional property of being incrementally maintainable. A detailed experimental evaluation of our approach with synthetic and real datasets demonstrates the applicability and efficiency of our techniques.

I. INTRODUCTION

Weblogs (or blogs) collectively represent a rich source of information on different aspects of a person's life, but more importantly on a myriad of different topics, ranging from politics and health to product reviews and issues of everyday life. As diversity is a natural feature of many areas which include a social aspect, the same is true in the context of weblogs. Users not only post their information, but also express their opinions on the topics discussed.

It is now becoming evident that the views expressed in blogs can be influential to readers in forming their opinions on some topic [19]. Similarly, the opinions recorded in blogs are an important factor taken into consideration by product vendors [18] and policy makers [28]. There is also evidence that this process has significant economic effects [3], [4], [8].

In this paper, we propose to perform an aggregated analysis of blogs in order to identify interesting patterns and trends related to the opinions expressed in these blogs. In particular, the focus is on automatic discovery of topics for which different opinions have been expressed across space and time. Consider the following motivating scenarios.

Health: There exists a substantial medical blogging community, which is composed from both trained and certified physicians (e.g., medical doctors) and individuals from the general population (e.g., patients, or relatives of patients). The blogs written by physicians allow patients and doctors alike to form an idea about what these professionals believe for some of the current topics related to health, and also how these views evolve over time. In contrast, the personal perspective of patient blogs allows physicians to learn about the mental, emotional and physical state of people living with certain medical conditions and how these change over time.

Politics: Blogs on various aspects of politics cover the entire spectrum of interested parties: from simple citizens expressing their opinions on everyday issues, to politicians using this medium in order to communicate their ideas (as was best exemplified during the last USA elections), and from journalists criticizing the government to the government itself (e.g., see the blog page of the White House [1]). It is to the benefit of all the parties mentioned above to follow the opinions that are expressed on a variety of topics in these blogs, and to be able to identify how these opinions or public sentiment change and evolve across time.

In both scenarios, one of the most intriguing aspects worth of investigating further is when the opinion of an individual or a group of people on a specific topic changes from positive to negative, or vice-versa. Examples of such cases are the opinion of doctors on a particular medical treatment, or the opinion of citizens on a particular government policy. We are interested in identifying these situations, when opinions change drastically, in a way that they become contradictory either with the prevalent opinions on the same topic at an earlier time period, or with the opinions of other groups of people (as expressed through their blog posts) in the same time frame. These contradictions are important, because - following our examples - they may signify a change of mind in the way a certain disease is treated, or may indicate a change of direction of the government with respect to some political issue.

In each case, we would like to be able to identify, record, and track such changes of sentiments and contradictions. The techniques we describe in this paper are focused on the tasks of extracting opinions from weblogs, identifying contradicting opinions, and providing an efficient and scalable way for managing this information, which can be the basis for analyzing the evolution of these opinions and contradictions.

Extracting Opinions: Opinions are reflected in blogs through the use of positively and negatively charged words. We introduce a two-step approach. In the first step, we employ Latent Dirichlet Allocation to analyze the sentences in the blogs, and assign them to topics. Subsequently, we calculate a continuous sentiment value for each topic in each blog, which expresses the opinion of the author on the corresponding topics.

Identifying Contradictions: While opinions can be summarized in many ways, previous work has focused on the opinion extraction step, not on the comparative study of these opinions. Even when different opinions are put together [24], [7], this is done primarily at the user interface level. In contrast, we are proposing a systematic approach on aggregating opinions with respect to some topic, and on identifying contradictions across different users. In this work, we describe mechanisms for effectively summarizing the opinions expressed in blogs. These summaries subsequently lead to a reliable and computationally efficient method for analyzing contradictions under different time granularities. Our approach also allows the identification of two different types of contradictions[1], namely, overlapping contradicting opinions (simultaneous contradiction), and opinions that shift over time (collective change of sentiment).

Managing Opinion Evolution: After having in place a method for aggregating opinions that can still preserve the information relevant to contradiction detection, an important problem is how to effectively use this information in order to analyze the evolution of opinions and contradictions over time. To this effect, we describe data structures that effectively summarize the contradictions over time, and can be efficiently queried to reveal interesting contradictions for different time intervals and time granularities. As shown in the experimental evaluation, these structures are scalable and incrementally updatable, when new blogs come in the system.

The contributions we make in this paper can be summarized as follows.

- We introduce and evaluate an approach for opinion analysis that identifies opinions per topic in a more granular manner (i.e., based on analysis of individual sentences) than previous approaches.
- We propose a systematic and effective mechanism for aggregating and organizing opinions over time and across blogs.
- We define two different types of contradictions, that is, simultaneous contradictions and collective change of sentiment, and we show how the above mechanism can be used to detect these contradictions in large collections of data.
- We describe incrementally maintainable data structures that can efficiently use the opinion summaries to answer *adhoc* queries on contradictions, on different time intervals and time granularities.
- Finally, we experimentally evaluate our techniques using synthetic and real datasets. The results demonstrate the validity of our approach.

[1]We formally define the different types of contradictions in Section III.

The rest of this paper is organized as follows. Section II gives an overview on work related to our topic. Then, the problems to be solved is described in more detail (Section III), followed by the description of our approach to solve these problems in Sections IV and V. The experiments and results are described in Section VII. We conclude and discuss directions for future work in Section VIII.

II. Related Work

In the following paragraphs, we discuss in some detail the problems of topic identification, opinion and sentiment extraction, and contradiction analysis. However, we should emphasize on the fact, that the problem of efficiency and scalability of opinion mining has not been studied well so far in the publications known to the authors.

A. Topic Identification

In this paper, blog contradictions are considered at topic-level, i.e., topics per blog post need to be discovered. To solve this task, different topic representation and detection methods are available, such as clustering of documents based on extracted keywords [35] or filtering of documents using networks of relations between tags [31]. The TopCat system [11] exploits natural language processing techniques to identify key entities in texts and then forms clusters with a hypergraph partitioning scheme. Substitution of topic identification with a lexicon look-up to determine product names, person names and the like as topics within the opinion mining task has been proven successful for processing specifically product or movie reviews [20], [33]. In addition, most of the existing research determines topics at document-level. Since we analyze sentiments on sentence-level, our approach also determines topics at this level.

The most relevant approach to our work is the work of Mei et al. [26], who propose a probabilistic topic sentiment model. Our approach differs in that sentiment calculation and topic detection are performed successively. Topic models are used to identify topics at sentence-level and the sentiment is represented on a continuous scale.

B. Opinion Mining and Sentiment Extraction

In existing research work, sentiment analysis is mostly considered as two- or three-class classification problem, distinguishing between *positive* or *negative* (or *neutral*) texts. Different lexical- and machine-learning approaches have been developed [30], e.g., using corpus statistics [36] or linguistic tools like WordNet [21]. The algorithms were mainly applied to movie [2] or product reviews [12].

Our approach goes beyond the classical classification problem and tries to assign a continuous value to a sentence reflecting the expressed opinion. The sentiment analysis task considered in this paper is most similar to the rating inference task in which the class labels are scalar ratings such as 1 to 5 "stars" representing the polarity of an opinion. Rating inference tasks were by now considered at document level [29] or on product feature-level [23], [32]. Pang and Lee [29] apply metric labeling to assign a value of a rating scale while Shimada and Endo [32] use frequency of words as classification features. Ku et al. [22] determine polarity scores between -1 and 1 indicating the polarity and the strength of a word. For this purpose, they calculate the frequency of single characters in positive and negative words of their opinion dictionary.

In contrast to existing rating inference approaches, our algorithm assigns a continuous value to each sentence or topic. Therefore, this task cannot be considered as multi-class classification problem. In contrast, our approach will rely on SentiWordNet scores as attributes for sentiment calculation. SentiWordNet [15] provides for each synset of WordNet[2] a triple of polarity scores (positivity, negativity and objectivity) whose values sum up to 1. It has been created automatically by means of a combination of linguistic and statistic classifiers and consists of around 207000 word-sense pairs or 117660 synsets. Existing work exploits this resource mainly for identification of opinionated words [14], [16].

C. Contradiction Analysis

A traditional approach in obtaining trends for popular items in blogosphere is to track user support for a set of popular keywords, i.e., measuring the frequency of keywords. Glance et al. describe BlogPulse [17], a system for identifying trends in weblog entries. This method uses frequency as a measure of popularity and relevance, but does not focus on how opinions may vary. Chi et al. [9] introduce a Singular Value Decomposition method for the analysis of trends in topic popularity across time. Some research work also examines how sentiments in blog entries of a single user change over time [25]. The problem of identifying and analyzing opinions has also been studied in the context of social networks. A recent study [10] examines how communities in blogosphere transit between high- and low-entropy states across time, incorporating sentiment extraction. Varlamis et al. [34] propose clustering accuracy as an indicator of blogosphere opinion convergence.

Closer to our work is the analysis of opinions expressed about commercial products, which has attracted particular attention in the research community. Morinaga et al. [27] describe a system for mining the reputation of products in the web. A similar approach is proposed by the Opinion Observer system [24] that focuses on summarizing the strengths and weaknesses of a particular product. Even though the above studies consider both positive and negative opinions, they do not aggregate them. In our approach, we describe an effective way for performing this aggregation, which leads to more insights into user opinions.

Chen et al. study precisely the problem of conflicting opinions [7] on a corpus of book reviews, which they classify as positive and negative. Their main goal is to identify the most predictive terms for the above classification task, and visualize the results for manual inspection. In contrast, we propose a systematic and automated way of performing opinion aggregation, revealing contradictions, and analyzing the evolution of these contradictions over time.

III. Problem Formulation

The problem we want to solve in this paper is to detect contradicting opinions on certain topics and to analyze their evolution across time in the blogosphere. In the rest of this section, we elaborate on these issues, and formally define the problems we address in this study.

A. Definition of Terms

Usually, a particular blog covers some general topic (e.g., health, politics) and has a tendency to publish more posts about one topic than another. Yet, within a blog post, the author may discuss several specific topics.

Defninition 1 (Blog Post Topic): A topic T is a named entity, event or abstract concept that is described in a blog post, P. We refer to all the topics contained in a single post as P *topics*, $\mathcal{T}^P$. Similarly, the blog posts that refer to a specific topic T are the T *posts*, $\mathcal{P}^T$.

For each of the topics discussed in a blog post, we wish to identify the author's opinion or sentiment towards it. In this study, we restrict ourselves to identifying and recording the *polarity* of these sentiments, which we represent as numbers. In addition to computing the sentiment polarity on a particular topic given an individual reference to it, we also need to compute the polarity on that topic aggregated over multiple posts (that may span different authors, as well as time periods). In the following, we refer to sentiment polarity simply as *sentiment*, and to the polarity of sentiments aggregated over a collection of posts as *topic sentiment*[3].

Defninition 2 (Sentiment): The sentiment S on topic T in a post P is a real number in the range $[-2, 2]$ that expresses the author's opinion on T. Negative values indicate negative opinions and positive values represent positive opinions.

Defninition 3 (Topic Sentiment): The Topic Sentiment S^T of a collection of posts $\mathcal{P}^T$, which are published within some predefined time window w on topic T, is defined as the aggregated value of the sentiments expressed in $\mathcal{P}^T$ with respect to T.

In this work, we use the range of $[-2, 2]$ to represent sentiment values, though, in principle any other range could be used as well. As will become evident later on, expressing sentiments using a continuous range of values gives us flexibility in aggregating and analyzing them.

We now turn our attention to the issue of comparing the sentiment values of different collections of posts.

[2]http://wordnet. princeton.edu/

[3]For the rest of this document we will use the terms *sentiment* and *opinion* interchangeably.

Definition 4 (Simultaneous Contradiction): In a collection $\mathcal{P}^T$ of posts talking about topic T, the topic T is considered contradictory, if there exist two groups of posts $\mathcal{P}_1^T, \mathcal{P}_2^T \subset \mathcal{P}^T$ such that the sentiment S_1 of $\mathcal{P}_1^T$ is very different to the sentiment S_2 of $\mathcal{P}_2^T$.

In the above definition, we purposely not specify exactly what it means for a sentiment value to be very different from another one. This definition can lead to different implementations, and each one of those will have a slightly different interpretation of the notion of contradiction. We believe that our definition captures the essence of contradiction, without trying to impose any of the particular interpretations. Though, later on (in section V) we propose a specific method for computing contradictions, which incorporates many desirable properties.

Another interesting situation arises when the majority of posts within some time interval exhibits a positive (negative) sentiment on a particular topic, and this time interval is followed by another one, where the majority of posts exhibits a negative (positive) sentiment on the same topic. Such time intervals, that contain a change of topic sentiment, can also be identified as contradictory, but with a special type of contradiction, which we call *Collective Change of Sentiment* (or simply, Change of Sentiment).

Definition 5 (Collective Change of Sentiment): We have a change of sentiment for topic T, at time t, when the following condition is satisfied: $\exists$ time interval τ : $\forall \epsilon \leq \tau : S^T(t-\epsilon)S^T(t+\epsilon) < 0$.

B. Definition of Problems

In order to detect contradicting opinions in collections of posts, we first need to determine all the different topics that appear in the posts, and calculate the sentiment of these topics.

Problem 1 (Topic Identification): Identify a set $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ of topics of interest that are discussed in the set $\mathcal{P} = \{P_1, P_2, \ldots, P_i\}$ of blog posts.

Problem 2 (Sentiment Extraction): For a topic $T \in \mathcal{T}^P$ in blog post P, we identify the sentiment S that has been expressed by the author on T in P.

Subsequently, we can detect the contradictions that appear in the dataset.

Problem 3 (Topic Contradictions): For a given topic T, identify the time windows w, contained in a specific time interval, τ, where a simultaneous contradiction or a change of sentiment occurs for T, with contradiction values above some threshold.

Problem 4 (Time Interval Contradictions): For a given time interval τ, identify the topics and time windows, within τ, for which these topics have a simultaneous contradiction or change of opinion above some threshold.

The time interval, τ, is user-defined, while the time windows, w, conform to an *apriori* segmentation of time (e.g., in weeks, months, etc.). As we will discuss later, the threshold can either be user-defined, or automatically determined in an adaptive fashion, based on the data under consideration.

The approach we propose in this work is general, and can lead to solutions for several variations of the above problem, such as detecting the topics with the highest contradiction or the most frequently contradicting topics.

IV. Extracting Opinions

The algorithm for analyzing contradictions works in two steps: First, for each topic discussed in a blog post, a sentiment value is calculated. Then, the actual contradiction analysis takes place. The methods for topic detection and sentiment analysis work on sentence level. Their results are later aggregated to come up with topic-sentiment pairs for the most relevant topics within one post. The different methods are described in the following sections.

A. Identification of Topics

For identifying topics per sentence, we apply the Latent Dirichlet Allocation algorithm (LDA, [5]), which was initially implemented to cluster complete documents according to their topic. We extended the algorithm by a sentence detection algorithm to apply it to sentences that are then considered as input 'documents' for the LDA and by a sophisticated preprocessing.

At the beginning of the topic detection step, each post is splitted into sentences using the Stanford NLP Tagger[4]. Besides detecting sentences, this tool assigns parts of speech to each word of a sentence and determines the words' base form. The LDA clustering then exploits only morphologically normalized words. Synonyms are considered using a manually created list of synonyms. The LDA algorithm identifies topics along with their probabilities based on the vector representation of sentences which in our case only considers normalized words. Each sentence as a topic mixture and each word's creation is attributable to one of the sentence's topics. Therefore, a topic is described by a set of words derived from the documents where to each word a probability is assigned that indicates the relevance of this word for the topic. In this way, all topics are described by the same words, but with varying probability values for each word. The number of topics has to be fixed at the beginning of the clustering process. We ran the LDA with standard parameters for α and β as reported by Steyvers and others who found that $\alpha = \frac{50}{t}$ and $\beta = 0.01$ where t is the number of topics work well with many different text collections [6].

In order to exclude sentences without topical focus, our LDA modification considers only words for clustering when they occur in at least 15 sentences. The probability per topic and sentence calculated by LDA indicates to what degree the sentence belongs to the topic. In our approach, only topics are considered relevant with a probability larger than $\frac{1}{n}$ where n is the number of topic clusters to be determined.

For sentiment analysis described in the next section, we consider the top 3 words of detected topics to be most relevant for describing a topic. Nevertheless, it can occur, that none of

[4]http://nlp.stanford.edu/software/tagger.shtml

the top 3 topic words can be found within a sentence to which this topic has been assigned. The LDA algorithm is not looking for matching keywords, but it is creating a model that describes the topic. The benefit of these topic models is that the correct topic can be assigned even if no matching keyword occurs in the sentence, just by relying upon a larger set of words, or the context, respectively.

B. Identification of Opinions

For each (relevant) topic determined by the modified LDA algorithm (see above), a continuous value between -2 and 2 is assigned indicating the sentiment expressed regarding this topic in the sentence under consideration. SentiWordNet [15] already provides continuous values representing the polarity of single words. Thus, we decided to develop an approach based on this resource. Other existing approaches (as discussed in Section II-B) assign a numeric value or distinguish only between *positive* and *negative* texts and are therefore not directly applicable to our scenario.

1) Sentiment Calculation: The polarity scores provided by SentiWordNet can be used in two different ways. We propose a rule-based approach, but also a machine-learning based approach is studied.

The sentiment regarding a topic is determined based on relevant opinionated words. Words are considered relevant if they appear close to a topic term, i.e. within a distance of four words before and after the topic term. We are exploiting the top 3 topic words for this purpose. In the case where none of the top 3 topic words can be found in the sentence under consideration, all words of a sentence are considered for sentiment detection. Stop words are removed, the resulting words are stemmed and their polarity score triples are collected from SentiWordNet. These values are in turn averaged which results in one polarity score triple per sentence. By calculating the difference between positivity and negativity value of this triple, the final continuous topic-related sentiment value is determined. Since SentiWordNet scores are in range between 0 and 1, we use a scaling factor of 2 to receive values in our sentiment range. If the resulting value is smaller than -2 (or larger than 2), the polarity value is set to -2 (or 2). Objectivity values are not considered in this rule since we only want to account for opinionated topics. In this way, to each topic of a sentence a sentiment value between -2 and 2 is assigned.

We also considered a machine-learning based approach to determine the sentiment and exploit a feature set that has previously been used to classify complete texts as *positive* or *negative* [13]. It consists of the number of positive, negative and neutral words, the number of adjectives, verbs and nouns, as well as the SentiWordNet triples of the five most frequent terms. The SentiWordNet score triple per term (calculation corresponds to that in the rule-based approach) are exploited to count the *positive* and *negative* words within a sentence. If the positivity value of a term is larger than the negativity value, the word is considered to be *positive* and *negative* otherwise. If both values are equal, the word is considered to be *neutral*. The resulting feature set is used by Linear Regression models [37] that allow determining a continuous sentiment value.

2) Assigning Sentiments to Document Topics: The previously described steps provide for each sentence of a post sentence-topic-sentiment triples. The sentiment values of sentences with the same topic are averaged to determine one sentiment value for each topic of a post. The final output of the sentiment analysis step is a continuous polarity value between -2 and 2 for each topic of a post.

The main contribution of the sentiment analysis approach is determining the semantic orientation of a sentence towards a topic in a more fine-grained manner using SentiWordNet. We decided to determine topics and sentiments at sentence-level to be able to consider changes of sentiment within one post. It may occur that regarding one topic different opinions are expressed in different sections of the same post. So, we have to identify all the word expressing the opinion towards this topic. For example, there is a WebMD post, where the author states as a fact that there are discussions on over-prescription of a certain drug. The matching topic keyword occurs only in this sentence. In the other sentences, he collects arguments in favor and against this statement, but resists on repeating the relevant topic keywords. By considering sentiment per sentence and relating it to the topic, as it is proposed by our approach, we are able to detect these different opinions regarding the same topic and to aggregate them.

V. Identifying Contradictions

Based on the analysis described so far, we are now in position to detect the contradicting topics. In the following paragraphs, we first propose a novel contradiction measure, and then describe a simple, yet effective way of organizing the data to identify contradictions based on this measure.

A. A New Measure of Contradiction

Following Definition 3, the topic sentiment for topic T can be calculated as the mean value of the opinions of all the posts that mention T, $\mathcal{P}^T$: $S^T = \frac{1}{n}\sum_{i=1}^{n} S_i$, where n is the cardinality of $\mathcal{P}^T$. Then, a value of S^T close to zero implies a high level of contradiction.

A problem with the above way of calculating topic sentiment arises when there exists a large number of posts with very low sentiment values (i.e., values close to zero). In this case, the value of S^T will be drawn close to zero, without necessarily reflecting the true situation of the contradiction. Therefore, we suggest to additionally consider the variance of the sentiments along with their mean value.

Defninition 6 (Topic Sentiment Variance): In a collection $\mathcal{P}^T$ of posts talking about topic T, the topic sentiment variance V_S^T is defined as follows: $V_S^T = \frac{1}{n}\sum_{i=1}^{n}(S_i - S^T)^2$. According to the above definition, when there is a large uncertainty about the collective sentiment of a collection of posts on a particular topic, the topic sentiment variance is large as well.

Figure 1 shows two example sentiment distributions. Distribution A with S^T close to zero and a high variance indicates

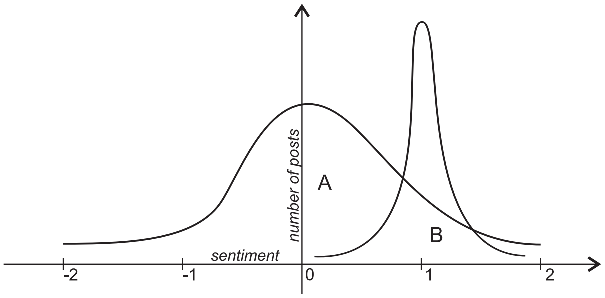

Fig. 1: Example of two possible sentiment distributions.

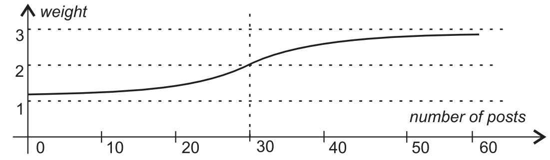

Fig. 2: The weight function used with number of posts in the criteria.

a very contradictive topic. Distribution B shows a far less contradictive topic with sentiment S^T in the positive range and low variance.

Evidently, we need to combine topic sentiment and topic sentiment variance in a single formula for computing contradictions. Assume that we want to look for contradictions in a shifting time window w. Without loss of generality, in this work we consider windows of a day, week, month, and year. For a particular topic T, the set of posts $\mathcal{P}^T$ will be restricted to those, that were posted within window w. We denote this set of T posts as $\mathcal{P}^T(w)$. Then, the contradiction value C^T can be computed as $C^T = \frac{V_S^T}{(S^T)^2}$, where S^T is squared so that its units are the same as the units of V_S^T.

B. Accounting for the Number of Posts

This formula captures the intuition that contradiction values should be higher for topics whose sentiment value is close to zero, and sentiment variance is large. Nevertheless, the contradiction values generated by this formula are unbounded (i.e., they can grow arbitrarily high as S^T approaches zero), and does not account for the number of posts in $\mathcal{P}^T(w)$. This latter point is important, because in the extreme where $\mathcal{P}^T(w)$ contains only two posts with opposite values, C^T will be very high, and will compare unfavorably to the contradiction value of a different set of T posts with a much higher cardinality.

Incorporating to the contradiction formula the observations made above, we propose the following formula for computing contradiction values:

$$C^T = \frac{V_S^T}{\vartheta + (S^T)^2} W \tag{1}$$

In the denominator, we add a small positive value, ϑ, which allows to limit the level of contradiction when $(S^T)^2$ is close to zero. Experimental results showed that any value between $0.05 - 0.1$ works well for all the synthetic and real datasets that we tried. In this study, we use a value of $\vartheta = 0.05$.

W is a weight function aiming to compensate the contradiction value for the varying number of posts that may be involved in the calculation of C^T. The weight function is defined as $W = 2 + \tanh\left(\frac{n}{10} - 3\right)$, where n is the cardinality of $\mathcal{P}^T(w)$. This weight function is a multiplicative factor in the range $[1, 3]$ (Figure 2 plots W as a function of n), which means that contradiction values fall within the interval $[0, 12/\vartheta]$. Using W we can effectively limit C^T when there is a minor number of posts, as well as when this same number of posts increases significantly. W is essentially a mechanism for lowering the contradiction value when the relevant posts are too few for the result to be credible, while not overcompensating in the presence of too many posts.

C. Applying the Contradiction Formula

Figure 3 demonstrates the operation of the proposed contradiction value function. The graph at the top (Figure 3(a)) shows a time series of synthetically generated sentiments for a period of 8000 time units. The dataset consists of 4000 normally distributed opinions with dispersion 0.5 and median following a custom trend. Additionally, we added 4000 points of normally distributed sentiments with dispersion 1 and median 0, acting like noise. Time stamps of all points follow the Poisson distribution with parameter $\lambda = 2$ time units. The bold line in this graph depicts the custom trend, showing an initial positive sentiment that later changes to negative (at time instance t_1). This behavior represents a change of sentiment. There is also a point around time instance t_2, where the sentiments are divided between positive and negative, a situation representing a simultaneous contradiction. Using this dataset, we verify the ability of the C^T function to capture the planted contradictions.

An important component of C^T is the topic sentiment, S^T. As can be seen in Figure 3(b), S^T closely captures the aggregate trend of the raw sentiments. The following two graphs in the figure show the contradiction value, calculated using a sliding window of size 500 and 1000 time units. When we use a window of small size (Figure 3(c)), C^T correctly identifies the two contradictions at points t_1 and t_2, where the values of C^T are the largest. Using a larger window has a smoothing effect in the values of C^T (Figure 3(d)). Nevertheless, we can still identify long-lasting contradictions: In this case, the largest value of C^T occurs at time instance t_1, corresponding to a change of sentiment that manifests itself across the entire dataset. The above observations also indicate the value of examining contradictions using time windows of varying cardinality. In the following paragraph, we describe how this can be done efficiently.

VI. Storing and Indexing Contradiction Values

So far we have described a technique for processing weblogs to extract sentiments on various topics, and subsequently to use this information in order to identify contradictions. We now turn our attention to the problem of organizing all these data in

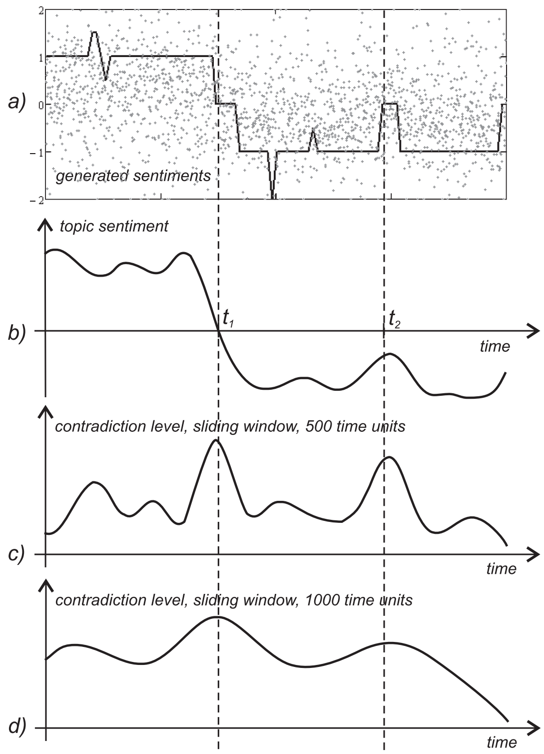

Fig. 3: Example of contradiction values computed from a synthetic dataset with two planted contradictions.

a way that will allow the efficient detection of contradictions in large collections of data that span very long time intervals.

An important observation is that the Formula 1 that calculates the contradiction values is based on the mean and variance of the topic sentiment. Remember that topic sentiment is calculated as $S^T = \frac{1}{n}\sum_{i=1}^{n} S_i$ (where n is the number of posts published on topic T in a specific time window). The topic sentiment variance can be written as $V_S^T = \frac{1}{n}\sum_{i=1}^{n}(S_i - S^T)^2 = \frac{1}{n}\sum_{i=1}^{n}(S_i)^2 - (S^T)^2$. We now define the first- and second-order moments of the topic sentiment as $M_1 = \sum_{i=1}^{n} S_i$ and $M_2 = \sum_{i=1}^{n}(S_i)^2$, respectively. Based on the above discussion, and using the sums M_1 and M_2, we can rewrite Formula 1 as follows:

$$C^T = \frac{nM_2 - M_1^2}{\vartheta n^2 + M_1^2} W^T \tag{2}$$

The above form of the contradiction values formula gives us additional flexibility, since we can now compute the contradiction of a large time window by composing the corresponding values from the smaller windows contained in the large one. We can therefore build data structures that take advantage of this property.

In the next paragraphs, we describe such a data structure, and we show how it can be used to identify contradictions. We also demonstrate that it can be easily maintained in an incremental fashion when new posts are added in the system.

A. *Contradiction Tree*

We introduce the Contradiction Tree (CTree) for managing the information on sentiments and contradictions. The CTree

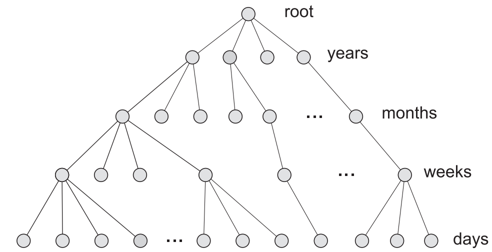

Fig. 4: Logical representation of the Contradiction Tree.

is organized around the sentiment moments, M_1 and M_2, and a hierarchical segmentation of time, as outlined in Figure 4. In this example, the time windows are organized on days, weeks, months, and years (though, other hierarchical time decompositions are applicable as well). Using this kind of structure, we can answer queries on *adhoc* time intervals, by dynamically computing the contradiction values based on Formula 2. In the following, we will refer to the levels of the CTree as the different *granularities* of the time decomposition, the root node having granularity 0.

Each node in the CTree corresponds to a time window, and summarizes information for all posts, whose timestamp is contained in this time window. The internal structure of the CTree nodes is illustrated in Figure 5. As the figure shows, a CTree node stores the following information: (a) for each topic, the topic id, tid, the number of posts, n, on this topic that fall in the time window represented by the node (we only store information for topics when $n > 0$), and the sentiment moments, M_1 and M_2; (b) pointers to the children nodes (black dots); and (c) pointers to adjacent nodes, *prev* and *next* of the same level (black diamonds). The adjacent node pointers are used to allow fast sequential access to neighboring nodes in the same time granularity.

In our implementation, we assume that each node fits in a single disk page. This translates to each node being able to hold information for 250 different topics (for our implementation). In the case where a node cannot fit all relevant topics, we can use additional storage, referenced by a special pointer in the CTree node (represented as a white dot in Figure 5). This solution allows us to accommodate a large number of topics at a small additional cost. Note that we can significantly reduce the expected cost of accessing this additional storage, by arranging the topics in a way that the most popular ones are located in the original node. For the purposes of this work we do not pursue this direction any further. Though, in the evaluation of our approach we report results with experiments that use this kind of additional storage.

B. *Querying the Contradiction Tree*

When trying to detect contradictions, we would like to identify those that have a contradiction value above some threshold. The intuition is that these contradictions are going to be more interesting than the rest in the same time interval. An obvious solution in this case is to define some fixed threshold,

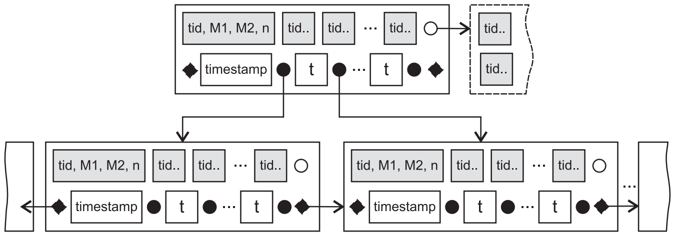

Fig. 5: Internal structure of the Contradiction Tree nodes.

ρ, and only report the contradictions above this threshold. We refer to this solution as *fixed threshold*.

However, by adopting the above solution, we cannot normalize the threshold to better fit the nature of the data within each time window (that may vary over time and across topics). In order to address this problem, we propose an *adaptive threshold* technique, which computes a different threshold for each topic and time window as follows. The adaptive threshold ϱ_w^T for a topic T in time window w is based on the contradiction value $C_{w_p}^T$ that has been calculated for T in the parent time window of w, w_p, and is defined as $\varrho_w^T = \mu C_{w_p}^T$, $0 < \mu < 1$. In our experience with real datasets, μ values between $0.5 - 0.7$ work well. In this work, we use $\mu = 0.6$.

Note that we cannot achieve the same result by using *top-k* queries (though, they can be complementary to our approach). The reason is that adaptive threshold does not impose a strict limit on the number of contradictions in the result, and can thus report the entire set of interesting contradictions within some time interval.

We are now ready to present the algorithm we use to solve the Topic Contradictions Problem, using time windows of a given granularity. Figure 1 outlines the algorithm that uses the adaptive threshold. The algorithm needs a single pass over the collection of pages of the specified granularity, l, that fall inside the time interval, τ of the query. In line 6, we check if a contradiction value (for a specific topic and time window) is above the adaptive threshold. Note that contradiction values, C^T are computed from the information stored in the node using Formula 2. The type of contradiction is identified in lines 7-9, by comparing signs of sentiments for adjacent nodes. In our implementation, we additionally do not visit children nodes whose parents are not contradictory (we omit this detail from the algorithm for ease of presentation).

The time complexity of this algorithm linearly depends on the number of nodes accessed to identify contradictions. This number, in turn, depends on the size of the time interval, τ, the size of the time windows of the chosen granularity, $|w_l|$. It also depends on the number of topics that are relevant to the time windows of granularity l, which in the worst case is the number of all topics, $|T|$. Therefore, the time complexity is $O(\frac{\tau}{|w_l|}\frac{|T|}{h})$, where h is the maximum number of topics that a single node can hold.

Solving the Time Interval Contradictions Problem is a

Algorithm 1: Identify contradictions

Input : Topic T, Time interval τ, Granularity l
Output: List of contradictions $\mathcal{C} = \{(time\ window,\ contradiction\ value,\ type)\}$
1 Set r to be the node at granularity $l-1$ containing the left endpoint of τ ;
2 Set output contradictions $\mathcal{C} = \emptyset$;
3 **repeat**
4 **forall** *nodes* $r_i \in r.childNode$ **do**
5 **if** $r_i.timeWindow \in \tau$ **then**
6 **if** $r_i.C^T > \mu \times r.C^T$ **then**
7 **if** $r_{i-1}.S^T \times r_i.S^T \leq 0$ **then**
8 *type* = "change of opinion";
9 **else** *type* = "simultaneous contradiction";
10 $\mathcal{C} = \mathcal{C} \bigcup (r_i.timeWindow, r_i.C^T, type)$;
11 $counter^T$ += 1;
12 **end**
13 **end**
14 **end**
15 $r = r.next$;
16 **until** r *is null or* $r.timeWindow.end \geq \tau.end$;
17 Arrange $\mathcal{C}$ by $counter^T$ or by C^T;

simple generalization of the same algorithm, where within each node we iterate over the topics stored in that node.

C. Updating the Contradiction Tree

As discussed earlier, the nature of the contradiction function (Formula 2) and the CTree nodes allows us to incrementally maintain the CTree in the presence of updates. When new blogs or individual posts are analyzed, their contribution to the contradiction of the corresponding topics and time windows in the CTree can be easily taken into account by updating the set of relevant $\{n, M_1, M_2\}$ values in the nodes of the tree.

Figure 2 shows the outline of this algorithm. When new posts arrive, as a preprocessing step, they are aggregated in time windows of the finer granularity of the CTree, w_j by computing their count, as well as the topic sentiment moments M_1^{upd} and M_2^{upd} for each topic. Then, these aggregate values are used to update the counts and topic sentiment moments of all CTree nodes whose time window contains w_j.

The update cost for each batch of aggregated posts depends on the depth of the CTree, d, and the number of topics, $|T|$ (in the worst case), that participate in the time windows relevant to the update. Thus, the complexity can be expressed as $O(d\frac{|T|}{h})$, where h is the maximum number of topics that a single node can hold.

VII. Experimental Evaluation

A. Experimental Setup and Datasets

The performance evaluation was conducted on a desktop computer with Intel Core 2 Duo CPU running at 2.53 GHz, 3Gb of RAM, and a 250Gb Samsung HDD HD251HJ, with Windows Vista Enterprise operating system. Our algorithms were implemented in Java and executed using Java JRE 1.6.13.

1) Description of Datasets: In our work we used the following real and synthetic datasets.

Synthetic Data: In order to evaluate the scalability of our solution, we generated a synthetic dataset by summarizing

Algorithm 2: Contradiction Tree update

Input : Topic T, sentiments $\mathcal{S} = \{S_i\}$, timestamps $\{t_i\}$
Output: Updated values
define `update` as a vector: (time interval τ, int n, float M_1, float M_2);
define `updateset` *upd* as a set {} of `update` vectors;
Aggr. sentiments for each smallest time interval τ: $\mathcal{S}_\tau = \{S_i | t_i \in \tau\}$
using the lowest granularity into a set of `update` vectors: Set upd =
$\{(\tau, n = |S_\tau|, M_1^{upd} = \sum_{S_\tau} S_i, M_2^{upd} = \sum_{S_\tau} (S_i)^2)\}$;
call `UpdatePage(`$rootPage$`,` upd`)`;
————————————————————————
function `UpdatePage(page` r`, updateset` upd`)`;
if $r.childPages \neq \emptyset$ **then**
    Set `update` $updResult = (upd.\tau, 0, 0, 0)$;
    forall *page* $rChild \in r.childPages$ **do**
        Set `updateset` $updChild = \emptyset$;
        forall *update* $u \in upd$ **do**
            if $u.\tau \in rChild.\tau$ **then** Add the update:
            $updChild = updChild \bigcup u$;
        end
        $updResult$ += `UpdatePage(`$rChild$, $updChild$`)`;
    end
else Aggregate input `updateset`: $updResult = \sum_{i=1}^{|upd|} upd_i$;
For a topic T in r, update the values:
Add the $updResult$ to $(r.n^T, r.M_1^T, r.M_2^T)$;
return $updResult$;

Attribute Name	Description
topicId	Topic identifier
timeBegin	Timestamp of the time window start
timeEnd	Timestamp of the time window end
granularity	A level of granularity
n	The number of posts within interval
M1	First-order moment of sentiments
M2	Second-order moment of sentiments

TABLE I: A schema for the table containing summary values.

approximately 80 million sentiments for 10,000 topics over a time interval of 4 years. The timestamp and sentiment values were drawn from a uniform distribution.

Real Data: Our algorithms are applied to a data set of health-related weblog posts from WebMD and a dataset with comments on postings from Slashdot (http://slashdot.org).

We crawled 28 health-related blogs with 2,405 posts covering 4 years (January 2005 to January 2009) from the WebMD webpage (http://blogs.webmd.com/). These posts are written by health care professionals and report on certain health topics such as disorders (e.g., *sleep disorders, asthma, anxiety*) or certain treatments (e.g., *cancer treatments*, *cosmetic surgery*).

Slashdot is a popular website for people interested in reading and discussing about technology and its ramifications, and includes posts, as well as comments on these posts. In this study, we used a dataset provided for the CAW2 workshop (http://caw2.barcelonamedia.org/) that contains about 140,000 comments under 496 articles, covering the time period from August 2005 to September 2006.

2) Baseline Solution: In order to compare the efficiency of the CTree, we also implemented our algorithms in a relational database, using the schema shown in Table I.

The database we used was IBM DB2 Express-C 9.5.2, with indices on the first four columns, and logging and transactions functionality turned off. Since DB2 uses a query cache, we averaged execution results among several runs, except the first one. Answering queries for the first time was on average 10 times slower than subsequent executions.

correct	Text
Topic ”back pain arthritis”	
yes	I don’t mind seeing patients with back pain.
yes	I have my daily aches and pains related to arthritis (a family legacy).
no	The last time this happened, I pushed to get back to work.
Topic ”government law federal”	
yes	This is a right only as long as it’s backed up by the power of the government.
yes	Unfortunately, the Commonwealth of Virginia has taken the exact opposite tact.
no	Sounds crazy that they’d agree to sign a contract like that.

TABLE II: Example sentences with detected topics.

B. Topic and Sentiment Extraction

1) Evaluation of Topic Detection: For evaluating the quality of the LDA on sentence-level, two people were asked to evaluate the top 3 topic keywords for the relevant topics determined by the LDA algorithm for 500 sentences of the Slashdot data set. When LDA clustered the sentences into 20 topics, the annotators marked topic words of only 21% of the sentences as correct. For the other sentences none of the suggested topic terms were relevant. Significantly better results of 54% were achieved when we increased the number of topics to 200 topics used by LDA.

In Table II some example sentences with assigned topics are listed. It can be seen that some sentences are correctly assigned, some even if none of the top 3 topic keywords are contained (e.g., second sentence for the topic ’government law federal’). On the other hand, for some sentences assigned topics seem to be unrelated to the topic (e.g., negative example for topic ’back pain arthritis’). The latter shows that the algorithm fails, when words are used in a different context (e.g., *back* in *get back* instead of *back pain*).

2) Evaluation of Sentiment Analysis: The accuracy of the proposed approach for sentiment analysis is determined based on the 500 sentences from the Slashdot dataset that have been manually annotated by four persons. None of them is an author of this paper. The average of the values assigned to each sentence by the four annotators are used as ground truth. We consider two annotations (ground truth vs. automatic assignment) for the same sentence as an agreement, when their difference is smaller than 0.5. We also calculated the ’error’ which is the difference between two annotations for the same sentence. The ’mean absolute error’ is calculated by averaging the ’errors’ of all annotated sentences. T-Tests are made to ensure statistical significance of the results.

The rule-based and the machine-learning based approach achieve similar mean error rates of 0.366 and an accuracy of 72% which is slightly better than the agreement of the human annotators with the ground truth (between 54% and 70%). Values assigned by our approach differ from the ground truth

Senti-ment	Text
Examples with correct assignments	
1.6	Practitioners and patients alike swear by the effectiveness of particular healing methods, even where there may not be a scientific explanation of how they work or even empirical evidence that they do really work.
1.5	It's easy to find really good examples of sensible taxation in the US.
-2.0	Both of these nasty arachnids can cause a painful bite, tissue damage, and even death.
-1.5	Something went wrong during the anesthesia and her little, normal brain was irreversibly damaged.
Examples with wrong assignments	
0.6	I believe that information technology is important, but I think that MIT is just trying to get publicity, something the Media Lab specializes in (Added nasty putdown - the Media Lab doesn't do very good science or engineering in my opinion.).
1.5	Valentine's Day has become an event filled with pressure to love on demand – and that's the very antithesis of romance or good sex.

TABLE III: Example sentences with sentiments detected by the rule-based approach.

most when also the human annotators disagreed to a large extent.

In Table III, we report some examples of correct and wrong sentiment assignments. Often, the misclassified sentences require background knowledge to correctly decide for a sentiment value which is missing in our current approach. E.g., for interpreting the sentence *Many Canadians themselves leave the country in what the government refers to as a 'brain drain'.* correctly, background knowledge is necessary. The approach also fails, when rather neutral words are used to express a *positive* or *negative* opinion e.g., *Isn't Cambridge deliberately creating an opportunity for the Chinese government to prosecute them?*. The last sentence in the examples in Table III shows that the algorithm also fails when negativity is expressed very subtly.

A more comprehensive evaluation of the topic-sentiment analysis approach will be reported in the full version of this paper.

C. Contradictions

We now apply the introduced contradiction analysis approach to the WebMD and the Slashdot dataset.

In Figure 6, the top graph depicts the raw sentiment values for the topic "internet government control" (from the Slashdot dataset), for the time interval January to September 2006. The following graphs show the topic sentiment and variance (two middle graphs), and contradiction values (bottom graph) for the above topic and time interval. Contradiction values have been calculated using a time window of one day. Note that contradiction values are high for the time windows where topic sentiment is around zero and variance is high, which translates to a set of posts with highly diverse sentiments. These situations are not easy to identify with a quick visual inspection of the raw sentiments.

The analysis shows that in this time interval there are three major contradictions (marked 1-3 in the bottom graph

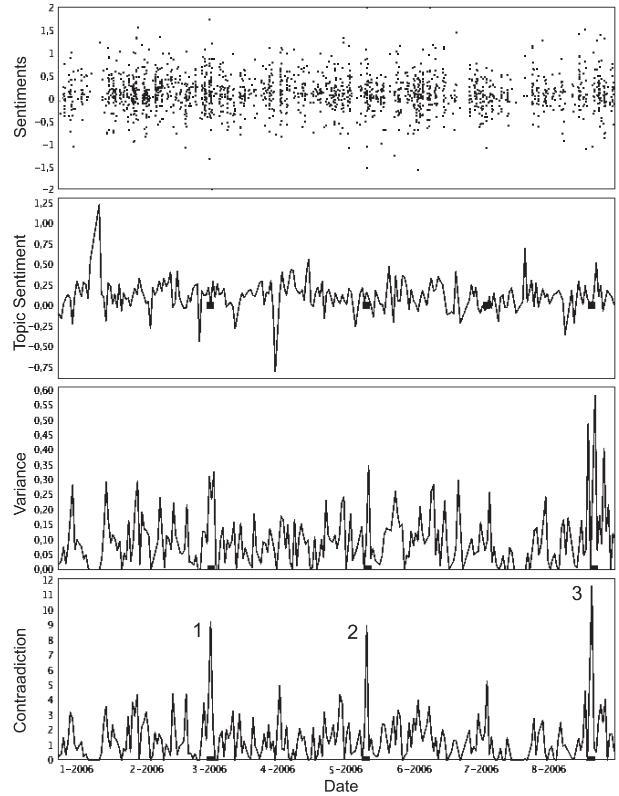

Fig. 6: Raw sentiment, topic sentiment, topic sentiment variance, and contradiction values for the topic "internet government control".

of Figure 6). All three contradictions discuss the pros and cons of a law that would give the government more power in controlling the internet traffic, especially personal correspondence. By taking a closer look at the corresponding weblog posts, we find out that the discussion around contradiction 1 is about web-related corporations operating in a monopolic or oligopolic environment, and how the above law would affect their operation and their customers. Contradiction 2 contains discussion of how this law can or cannot deter attacks from foreign countries, while contradiction 3 discusses the ways that this law will or will not affect national and foreign citizens.

Table IV shows extracts from two opposing posts that contributed to contradiction 2. In the same table, we also report additional examples of contradictions identified by our analysis. For the topic "iraq war", the related posts discuss pros and cons of the US strategy on this issue. The extracted posts correspond to a change of sentiment contradiction that our algorithm identified in the time period of May 2006. This particular change of sentiment was from positive to negative. Interestingly, it coincides with a surge of bomb attacks in Iraq, which claimed many US lives.

The next two examples of contradictions come from the WebMD dataset. In the first one, falling under the topic "adhd child", the corresponding posts discuss treatment of AD/HD (Attention-Deficit Hyperactivity Disorder, a commonly diag-

topic ”internet government control”, Slashdot (contradiction 2 in Figure 6)	
PRO	How about to make a positive impact on the world by gathering and protecting information to prevent terrorists from carrying out acts of violence and to stop hostile countries from threatening the security of the United States and its allies. Because that is what the NSA does!
CON	How do you want to block a top level domain? At the end, you'll find out that those sites will be accessed via the IP address. You're making inappropriate assumptions here.
topic ”iraq war”, Slashdot	
PRO	You are fortunate to live in a country unencumbered by an ongoing threat of terrorism and I respect your governments decision to oppose the U.S. attack in Iraq.
CON	Unfortunately, that happened to many Americans during the run-up to the ongoing war in Iraq. Most Americans didn't investigate the claims made by politicians and the media, and thus were ignorant to the fact that they were being seriously mislead.
topic ”adhd child”, WebMD	
PRO	I have seen antidepressants make a huge positive difference lifting a child's mood and improving the quality of his/her life.
CON	Stimulants treat symptoms of ADHD in a greater percentage of people than [Brand Name Drug], and often treat inattention and destructibility more robustly than [Brand Name Drug]. [Brand Name Drug] isn't safer than a stimulant, and if effective a stimulant alone would be a far better choice.
topic ”diet fat day”, WebMD	
PRO	Any decrease in breast cancer in the experimental group would be measurable by comparing the two groups.
CON	A low fat diet does not decrease the incidence of invasive breast cancer in post-menopausal women.

TABLE IV: Examples of contradicting posts.

nosed psychiatric disorder in children). One group of posts speaks in favor of a specific brand name drug, which is an antidepressant, while others indicate the disadvantages of this drug, and suggest a different drug. In the second example, under the topic “diet fat day”, a set of posts present results where low fat diets decreased the incidence of breast cancer, while others report the opposite.

Evidently, these are all very relevant discussions that express different points of view on the same topic, and having an automated way of identifying them can be very useful.

D. Scalability

We evaluate the scalability of the CTree for solving Problems 3 and 4, and compare its performance to a relational database implementation. Remember that in the Topic Contradiction problem we want to identify the contradictions and corresponding time windows of a single topic within some time interval, while in the Time Interval Contradictions problem we are interested in doing the same for all topics.

To test the performance of our solutions, we generated sets of 25 single-topic and all-topics queries (corresponding to the Topic and Time Interval Contradictions problems, respectively), with time intervals and topic ids drawn uniformly at random. In these experiments, we used 1,000 topics. We measured the time needed to execute these queries against the CTree and the database as a function of the time interval, τ, and the granularity of the time windows (Figure 7). We report results for both the fixed threshold and the adaptive threshold.

The adaptive threshold queries require in all cases more time since the threshold in this case has to be computed based on the contradiction value of the parent time window, which incurs more computation. This difference is pronounced for the database implementation, because it involves an extra join (for obtaining the parent time window). On the other hand, the same functionality in the CTree is achieved by following pointers, resulting in a minimal additional cost.

We observe that both single-topic and all-topics queries (see Figures 7(a-b)) scale linearly with the size of τ. This confirms our analytic results, and is explained by the fact that the queries have to return contradictions for all time windows (of a specific granularity) that are contained in τ. The CTree approach performs 1-4 orders of magnitude faster than the database, except for single-topic queries with fixed threshold. In this case, the database is able to use all its indices (i.e., on topic id, time windows, and granularity) to answer the queries, therefore, achieving very fast response times.

Figures 7(c-d) depict the time results when we vary the granularity of the time windows specified by the queries. Increasing the granularity translates to larger time windows (i.e., moving up in the time hierarchy) and a smaller number of time windows for the same time interval. Thus, response times get lower. Once again, we observe the same trends in the relative performance between CTree and database implementation as with the previous experiments on varying time intervals.

Finally, we measured the time needed to update the CTree and the database with information from new posts. The updates in the database were executed as batch updates, and logging was turned off. In Figure 8, we report the average time to perform 1,000 updates as a function of the number of topics. Each update operation corresponds to the update of a time window of the finest granularity (and consequently, of all its ancestors as well), or the creation of a new such window (and the update of its ancestors).

The graph shows that there is a linear dependency between the update cost and the number of topics in the system. As we discussed in Section VI, the increased cost for CTree comes from accessing additional nodes for each time window, when the number of topics do not fit in a single node. Nevertheless, CTree still performs 4 times faster than the database implementation.

VIII. Conclusions

In this paper, we study the problem of identifying contradictions in weblog posts. In the approach we propose, the posts are first processed in order to extract the topics mentioned in them, and then a sentiment value is assigned to each one of these topics. Subsequently, the sentiment values are aggregated for each topic and across different time windows, according to a novel function for computing contradiction values. These values are finally organized in a tree structure, which can be queried to report contradictions. The experimental evaluation, with synthetic and real datasets, demonstrates the usefulness of our approach and the efficiency of the proposed solution. In

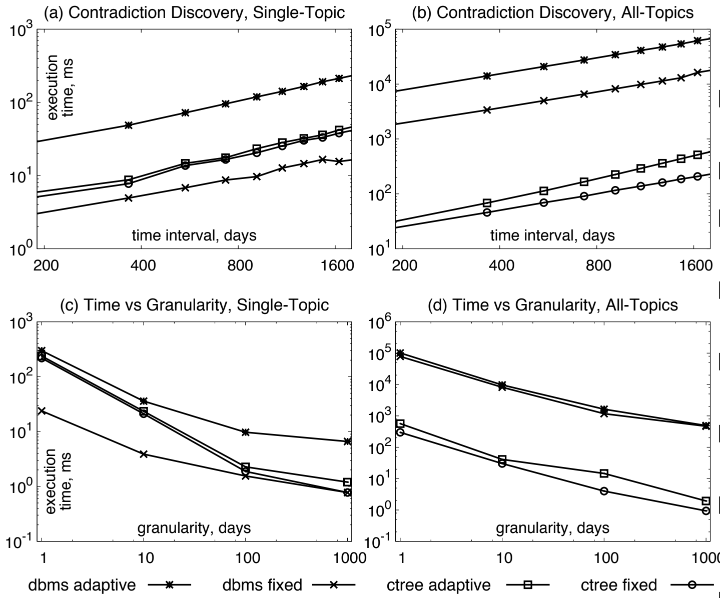

Fig. 7: Scalability of single-topic and all-topics queries.

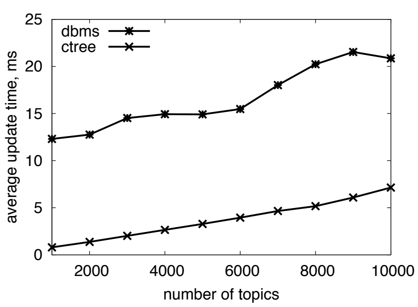

Fig. 8: Update time as a function of the number of topics.

future work, we plan to investigate more sophisticated ways in which to analyze the identified contradictions.

REFERENCES

[1] White House Briefing Room Blog Page. http://www.whitehouse.gov/blog/, 2009.

[2] M. Annett and G. Kondrak. A comparison of sentiment analysis techniques: Polarizing movie blogs. In *Canadian Conference on AI*, pages 25–35, 2008.

[3] W. Antweiler and M. Z. Frank. Is all that talk just noise? the information content of internet stock message boards. *Journal of Finance*, 59(3):1259–1294, 06 2004.

[4] N. Archak, A. Ghose, and P. G. Ipeirotis. Show me the money!: deriving the pricing power of product features by mining consumer reviews. In *KDD*, pages 56–65, 2007.

[5] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty. Latent dirichlet allocation. *JMLR*, 3, 2003.

[6] C. Chemudugunta, P. Smyth, and M. Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. In *NIPS*, pages 241–48, 2005.

[7] C. Chen, F. Ibekwe-SanJuan, E. SanJuan, and C. Weaver. Visual analysis of conflicting opinions. In *IEEE Symposium on Visual Analytics Science and Technology*, pages 59–66, 2006.

[8] J. A. Chevalier and D. Mayzlin. The effect of word of mouth on sales: Online book reviews. *Journal of Marketing Research*, 43(3):345–354, 2006.

[9] Y. Chi, B. L. Tseng, and J. Tatemura. Eigen-trend: trend analysis in the blogosphere based on singular value decompositions. In *CIKM*, pages 68–77, 2006.

[10] M. D. Choudhury, H. Sundaram, A. John, and D. D. Seligmann. Multi-scale characterization of social network dynamics in the blogosphere. In *CIKM*, pages 1515–1516, 2008.

[11] C. Clifton, R. Cooley, and J. Rennie. Topcat: Data mining for topic identification in a text corpus. In *TKDE 16(8)*, pages 949–964, 2004.

[12] K. Dave, S. Lawrence, and D. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *WWW*, 2003.

[13] K. Denecke. How to assess customer opinions beyond language barriers? In *Third International Conference on Digital Information Management, ICDIM*, pages 430–435, 2008.

[14] A. Devitt and K. Ahmad. Sentiment polarity identification in financial news: A cohesion-based approach. In *45th Annual Meeting of the Association of Computational Linguistics*, 2007.

[15] A. Esuli and F. Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. *Proc of LREC 2006 - 5th Conf on Language Resources and Evaluation*, 2006.

[16] A. Fahrni and M. Klenner. Old Wine or Warm Beer: Target-Specific Sentiment Analysis of Adjectives. In *Proc.of the Symposium on Affective Language in Human and Machine, AISB 2008 Convention*, pages 60 – 63, 2008.

[17] N. Glance, M. Hurst, and T. Tomokiyo. Automated trend discovery for weblogs. In *WWW*, 2004.

[18] T. Hoffman. Online Reputation Management is Hot - But is it Ethical? Computerworld, feb 2008.

[19] J. A. Horrigan. Online shopping. Pew Internet and American Life Project Report, 2008.

[20] M. Hu and B. Liu. Mining opinion features in customer reviews. In *AAAI*, pages 755–760, 2004.

[21] J. Kamps, M. Marx, R. Mokken, and M. de Rijke. Using wordnet to measure semantic orientations of adjectives. In *Proc LREC 2004*, 2004.

[22] L.-W. Ku, Y.-S. Lo, and H.-H. Chen. Using polarity scores of words for sentence-level opinion extraction. *Proceedings of NTCIR-6 Workshop Meeting*, pages 316–322, 2007.

[23] C. W.-K. Leung, S. C.-F. Chan, and F.-L. Chung. Integrating collaborative filtering and sentiment analysis: A rating inference approach. In *ECAI 2006 Workshop on Recommender Systems*, pages 62–66, 2006.

[24] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *WWW*, pages 342–351, New York, NY, USA, 2005. ACM.

[25] R. McArthur. Uncovering deep user context from blogs. In *AND*, pages 47–54, 2008.

[26] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW*, pages 171–180, New York, NY, USA, 2007. ACM.

[27] S. Morinaga, K. Yamanishi, K. Tateishi, and T. Fukushima. Mining product reputations on the web. In *KDD*, pages 341–349, 2002.

[28] T. Mullen and R. Malouf. A preliminary investigation into sentiment analysis of informal political discourse. In *AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*, 2006.

[29] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*, 2005.

[30] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[31] L. Qu, C. Müller, and I. Gurevych. Using tag semantic network for keyphrase extraction in blogs. In *CIKM*, pages 1381–1382, 2008.

[32] K. Shimada and T. Endo. Seeing several stars: A rating inference task for a document containing several evaluation criteria. In *PAKDD*, pages 1006–1014, 2008.

[33] V. Stoyanov and C. Cardie. Annotating topics of opinions. In *International Conference on Language Resources and Evaluation*, 2008.

[34] I. Varlamis, V. Vassalos, and A. Palaios. Monitoring the evolution of interests in the blogosphere. In *ICDE Workshops*, pages 513–518, 2008.

[35] C. Wartena and R. Brussee. Topic detection by clustering keywords. In *DEXA*, pages 54–58, 2008.

[36] J. Wiebe. Learning subjective adjectives from corpora. In *Proc 17th National Conf on AI and 12th Conf on Innovative Applications of AI*, 2000.

[37] I. Witten and E. Frank. *Data Mining. Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco, 2000.