# Managing Diverse Sentiments at Large Scale

Mikalai Tsytsarau and Themis Palpanas

**Abstract**—The large-scale aggregation and analysis of user opinions is becoming increasingly relevant to a variety of applications, from detecting social mood on some political topics to tracking their sentiment changes related to events. The analysis of diverse sentiments is another important application, which becomes possible based on the ability of modern methods to capture sentiment polarity on various topics with high precision and on the ever-growing scale. Therefore, there is a need for a scalable way of sentiment aggregation with respect to the time dimension, which stores enough information to preserve diversity, and which allows statistically accurate analysis of sentiment trends and opinion shifts. In this paper, we are focusing on the novel problem of aggregating diverse sentiments at a large scale, based on data sources that are continuously updated. First, we develop a theoretical framework that models sentiment diversity (contradiction) and defines two types of contradictions, depending on the distribution of sentiments over time. Second, we introduce novel measures that capture sentiment diversity from aggregated sentiment statistics. Third, we develop robust and scalable indexing and storage methods for diverse sentiments. Finally, we propose an adaptive approach for identifying contradictions at different time scales. The experimental evaluation demonstrates the effectiveness of the proposed method of capturing contradictions and its superiority over relational databases in real-world scenarios.

**Index Terms**—Sentiment aggregation, opinion mining, contradiction analysis

## 1 Introduction

During the recent years we have been witnessing the proliferation of online platforms that allow people to publish their opinions, such as microblogs, social networks, forums and others. They all represent a rich source of opinionated information on different topics, which can be analyzed and exploited in various applications and contexts. Large-scale sentiment analysis can be used, for example, to learn about customers attitude to a product or its features [1], to monitor sentiments across various demographic groups [2] or to reveal people's reaction to some event [3]. Such problems require scalable and robust analysis of big social data to produce a desired output, calling out for new methods that enable finer sentiment recognition as well as larger application scale [4].

The opinion can be either a definitive statement, e.g., "dress is *black and white*", or an evaluative statement, e.g., "this dress looks *nice*". In this work, we are interested in contradicting ones, i.e., those that have no sense together. For example, claims "dress is *black and white*" and "dress is *blue and gold*" as well as claims "this dress *looks nice*" and "this dress is *unfashionable*" cannot be both true when referring to the same dress, even coming from different authors. The latter example represents a contradiction between opinions of the evaluative type, which are called sentiments. Sentiments can be assigned a polarity score, ranging from pleasantness (positive) to unpleasantness (negative).

- *M. Tsytsarau is with DISI, University of Trento, Via Sommarive, 9, Trento, Italy. E-mail: mikalai.tsytsarau@gmail.com.*
- *T. Palpanas is with LIPADE, Paris Descartes University, Paris 75006, France. E-mail: themis@mi.parisdescartes.fr.*

*Manuscript received 23 Jan. 2015; revised 1 July 2016; accepted 19 July 2016. Date of publication 3 Aug. 2016; date of current version 3 Oct. 2016.*
*Recommended for acceptance by T. Li.*
*For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.*
*Digital Object Identifier no. 10.1109/TKDE.2016.2597848*

The problem of aggregating diverse sentiments (and detecting their contradiction) has been studied in the context of different research areas, from product review mining to information retrieval [5]. Recently proposed methods can compute average positive and negative sentiments expressed on some topic and extract a representative text summary of polar opinions on various aspects of that topic [1], [4], [6], [7]. However, the information contained in average sentiments may be incomplete. For example, if two opposite sentiment values are summed up, the result may have a neutral polarity. The information about either sentiment is then lost. On the other hand, representative expressions of opposite opinions are only capturing the meaning of contradiction, but not its level. Therefore, this problem essentially requires a consistent definition and new methods to deal with.

Fig. 1 demonstrates the example of aggregated diverse sentiments for some topic, and why it can be misleading to consider the simple average of sentiments. Here, we plot the intensity over time of the positive and negative sentiments, highlighting the time intervals with high sentiment diversity, such as simultaneous contradictions (1) and changes of sentiment (2). The net sentiment in both of the demonstrated cases is equal to zero, yet neutral sentiments do not even present in any of the highlighted groups.

In order to give a closer look at diversity of sentiments and understand how different sentiment distributions perceived by people, we performed a user study [8] using the three datasets of topic comments from diverse domains: drug ratings, YouTube, and Slashdot. We asked users to select from these discussions a few groups of consecutive comments, which were contradicting, and then manually annotated their sentiments. From these we obtained 64 sets of sentiments from strictly contradictory groups of comments, marked by the majority of users. In contrast to that, we also selected 64 consecutive sentiment groups, not marked as contradictory by any user. Sentiments of each group were represented by a histogram counting five bins:

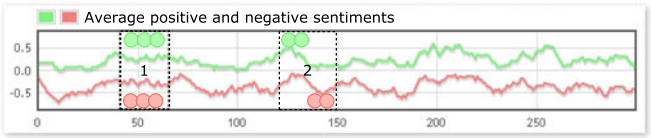



Fig. 1. Trends of diverse sentiments over time.

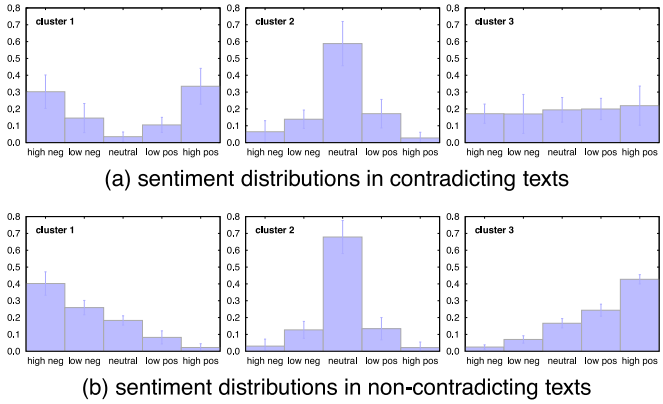
"high neg", "low neg", "neutral", "low pos", "high pos". Consequently, we applied *k-means* clustering with euclidean distance metric on contradicting and non-contradicting histograms, aggregating each collection into $k = 3$ clusters, visualised in Fig. 2.

In this figure, we observe that contradicting groups of texts (top) have nearly symmetrically balanced positive and negative sentiments, while non-contradicting ones (bottom) have either positive or negative deviations of sentiment. The only exception to the above statement is cluster 2 in Fig. 2b, which resembles cluster 2 in Fig. 2a, but contains 10 percent more neutral sentiments and is less polarised. Therefore, it is not possible to detect contradictions by only looking at the average sentiment or variance.

To address the above problems, we represent a method for aggregating diverse sentiments, which aims at supporting and facilitating large-scale sentiment analysis. First, we introduce a framework that theoretically establishes the concept of sentiment contradiction and addresses relevant problems of diverse sentiment aggregation [8]. Second, we develop a method which operates on continuous polarized sentiment values, allowing us to exploit different approaches for sentiment extraction, even multidimensional, which can be plugged in our framework. The use of only a few statistical aggregates for storing diverse sentiments allows our method to be extremely memory-efficient and scalable. Moreover, we apply adaptive regression and smart thresholding to deal with sentiment extraction noise and sentiment irregularity in such cases when sentiment data is scarce. The main contributions of this work can be summarized as follows.

- We formally define the notion of *opinion diversity*, introducing synchronous and asynchronous opinion contradiction types, and formulate relevant problems of contradiction detection.
- We present an approach, which solves the above problem for *sentiment contradictions* by using a novel sentiment diversity measure based on statistics of sentiment distribution.
- We describe a novel data structure, named CTree, which enables our approach to scale to very large data collections. It is incrementally maintained, and can outperform a relational DBMS implementation by up to three orders of magnitude.
- We experimentally evaluate our methods, demonstrating their effectiveness and scalability.

The remainder of this paper is structured as follows. In Sections 2 and 3 we discuss the related work and methods, and in Section 4 we formally define the problem. We present our approach for detecting contradictions in Section 5, and describe a scalable implementation in Section 6. The experimental evaluation is shown in Section 7. Finally, we conclude in Section 8.




(a) sentiment distributions in contradicting texts

(b) sentiment distributions in non-contradicting texts

Fig. 2. Observed sentiment distribution types.

## 2 Related Work

In this section, we briefly introduce the literature on sentiment and topic analysis, then focusing on the problem of contradictions, occurring with large scale sentiment analysis [1], [4], [5], [9].

### 2.1 Problems of Sentiment Analysis

To this end, sentiment analysis was mostly considered as two- or three-class opinion polarity classification problem, distinguishing between *positive*, *negative* or *neutral* texts [5]. Different lexical, statistical, semantic and machine-learning approaches have been developed for sentiment analysis and applied to various kinds of texts, from movie and product reviews to blog posts and tweets. However, the scientific frontier has recently shifted towards multimodal and semantics-based extraction of fine-grained sentiments [4], [9], [10]. Such methods aim at drilling-down on the expressed sentiments and at extracting their polarities with the maximum accuracy, depending on discussed topics, the surrounding context, the discourse structure and many other attributes, that are relevant to provide a complete micro-view of opined text. In this work, we are more interested to develop a macro-view on the sentiments coming from multiple texts and from different authors, in order to monitor changes of sentiments on a global scale. Therefore, we do not develop any new techniques for sentiment extraction, instead, concentrating on the problem of storing and analysing large volumes of polarised multi-dimensional numeric data.

### 2.2 Problems of Topic Tracking

An important requirement to capturing and tracking opinions is the ability to reliably detect and follow their topics. For product reviews and for large texts topics may include some general concepts mentioned only indirectly, like *'service quality'*, *'economic politics support'*, requiring latent topic modelling [11]. For shorter and more specific texts, like Twitter messages, topics are usually found as named entities, products, people and events, mentioned directly within sentiments. Therefore, topic-dependent sentiment analysis and aspect sentiment analysis already became standard tools for opinion extraction [1], [7]. Detecting topics in a large-scale scenario necessitates special methods that can deal with huge topical spaces and document volumes [12]. Moreover, sentiment shifts are often associated with emergent sub-topics, that demand special

foreground-background topic modelling, like in FB-LDA [13], or adaptive classifier features, like in TASC-t [14] to pinpoint the underlying changes. Still, there are many more relevant examples of topic and sentiment model adaptation to our problems. An interested reader can get familiar with relevant approaches for sentiment topic and aspect detection in a survey by Schouten and Frasincar [7], while our work concentrates mainly on managing numeric sentiments in order to detect their contradictions.

### 2.3 Problems of Contradiction Analysis

*Contradiction analysis* is a rather new research area, developed within linguistic analysis and finding its application to opinion mining. De Marneffe et al. [15] define contradiction as a situation where *'two sentences are extremely unlikely to be true together'*, and approach this problem using textual entailment principle, which employs linguistic processing. They also introduce a classification of contradictions consisting of seven types according to features that contribute to a contradiction (e.g., antonymy, negation, numeric mismatches). Since then, many other linguistic and semantic approaches for sentiment analysis were exploiting contradictions, albeit, requiring complex text processing.[1] Our approach is based on numeric sentiments and is intended for large-scale operation, where pairwise comparisons of texts, yet even any kind of linguistic analysis are not computationally efficient.

Sentiment time series may demonstrate outbursts of *diverse sentiments* and rapid *sentiment changes* (which are synchronous and asynchronous types of contradictions) that were recently studied in several publications addressing Twitter sentiments. Popescu and Pennacchiotti [16] propose a hybrid approach of detecting contradictions in Twitter, which is based on a machine learning classifier trained on a rich set of textual and statistical features. Although this improves the precision over purely textual and purely statistical feature sets, selecting the right combination of features requires numerous training and adaptation stages, especially for texts coming from different sources. Thelwall et al. [17] evaluate how twitter sentiment and its volume is changing before and after news events. By analyzing the peaks in sentiment, they show that the volume of negative sentiment is increasing just before an event, while there is an increase of positive sentiment at the event's peak intensity. Another observation is that the changes in sentiment are particularly small, making it necessary to apply more sophisticated methods capable of detecting them under high noise conditions, thus stressing the need for our work. Finally, Morales et al. [18] study sentiment distributions in Twitter during major events, and propose a new measure for sentiment polarisation based on the distance between mean values of positive and negative sentiment distributions and on their relative proportions, inspired by dipole moment. While this measure is effective for its purposes, it is relying on the entire distribution of sentiments for computing, requiring much more data to store and analyse, compared to our contradiction measure.

1. More recently, Xia et al. [10] proposed using only antonymy dictionary and negation method for enriching a training set for bag-of-word classifiers, achieving good results along with faster text processing.


Fig. 3. Workflow of diverse sentiment analysis.

Problems related to identification and analysis of contradictions have also been studied in the context of social networks and blogs. Dînşoreanu and Potolea [19] use semantic information to group users with similar sentiments and to detect opinions of individual users, which contradict with their community or with previously posted opinions. By analysing semantic relations of opinion targets and extrapolating community opinions to individual users, the authors made possible to retrieve more of potential contradictions. Choudhury et al. [20] examined sentiment biases in blogosphere communities, relying on an entropy measure as an indicator of the diversity in opinions. Clustering accuracy as an indicator of blogosphere topic convergence was proposed by Varlamis et al. [21]. Unlike clustering, our method allows storing data in an efficient and incrementally updatable manner, allowing ad-hoc queries.

Overall, opinion contradictions and diverse sentiment aggregation, as considered here, is a promising track for explorations, which can lead to interesting problem formulations and approaches. There are several such applications, which took off from the ideas and techniques described in this paper. An extended version of the proposed sentiment storage was used by [2] to monitor sentiments across tens of thousands of various demographic groups, automatically detecting their correlations and sentiment biases. Such kind of analysis wouldn't be possible without efficient time indexing and aggregation of sentiments. Also, automatic detection of significant contradictions and sentiment shifts despite noisy and irregular observations, discussed in this paper, helped to capture the reaction of social media to various kinds of news events [3], [22], revealing, for instance, different kinds of sentiment shifts for expected and unexpected events.

## 3 Topic and Sentiment Retrieval

In this paper we address the problems of efficient management of diverse sentiments and contradiction detection on specific topics from large-scale, noisy data streams. To tackle the large volume, irregularity, noise and other issues associated with big data, we propose a two-step approach to our problem, consisting of sentiment aggregation and the subsequent analysis, which follow the preliminary steps of topic and sentiment extraction, as demonstrated in Fig. 3.

Our processing starts with a data source, which outputs texts conveying opinion, that are either entire web documents (e.g., *blog posts, comments, tweets*), or some relevant parts of them (e.g., *sentences*) where an author discusses topics, which we want to identify in the first step, *Topic Extraction*. For each of these topics, we wish to extract the expressed opinions, in the second step, *Sentiment Extraction*. These steps can be accomplished using existing methods, or adaptations of existing methods. We refer to these steps as 'preprocessing' and briefly describe in the following how we have adapted them. The focus of our work is then on the subsequent two steps, namely, the aggregation of extracted sentiments and their analysis in order to detect contradictions.

We determine topics and opinion expressions at sentence-level to be able to capture fine-grained sentiments within their context in a text. Nevertheless, we consider average sentiments of the same topic, and obtain one sentiment value for each topic in a text. This is done to equalize the participation of each document in the aggregated sentiment and to prevent the argumentation within some documents from affecting the contradiction level. As an additional benefit, this step reduces the amount of data to process.

For the topic and sentiment assignment steps, we use the LK tool for fine-grained opinion analysis [23]. This tool achieves good results for opinion expressions detection and sentiment assignment by combining the two tasks and applying a re-ranking classifier to the output. Another feature of this tool is unsupervised dictionary-based sentiment assignment, which is rather useful for processing opinions for a variety of topics coming from different domains. For each document-topic pair we assign a continuous sentiment value in the range [-1;1], by averaging LK output sentiments: "high neg" ($-1.0$), "low neg" ($-0.5$), "neutral" (0), "low pos" (0.5), "high pos" (1.0).

Topic-sentiment pairs detected by LK (such as named entities, persons, events and concepts) are then filtered according to a pre-determined list of topics of interest, before adding them to our storage CTree. Therefore, every stored topic time series traces only sentiments addressed to a specific subject, making the analysis more consistent and reliable. In this paper, CTree is evaluated using large, but a fixed number of topics overall, which can be variable across different time intervals. That is, one time interval may contain sentiments for topics '2', '3', '5', and '6', whereas another only for '1', '3', and '4', not requiring any additional storage. Moreover, past and current time intervals at all levels of the tree remain fully updatable for newly-discovered topics, which can be traced in the root node. Nevertheless, for the lack of space, we leave topic tracking as a possible extension, and concentrate on modelling diverse sentiments.

## 4 Problem Definition

Following our introduction, we are ready to define contradicting opinions and opinion shifts (equivalently, contradicting sentiments and sentiment shifts), as described below.

**Definition 1 (Opinion).** *$O$ represents a personal non-ambiguous statement, claim or belief expressed by an author on topic $T$.*

**Definition 2 (Sentiment).** *The sentiment $S$ with respect to a topic $T$ is a vector that indicates the polarity of expressed evaluative opinion along basic emotional dimensions [24], such as Joy $\Leftrightarrow$ Sadness, Acceptance $\Leftrightarrow$ Disgust, Anticipation $\Leftrightarrow$ Surprise, and Fear $\Leftrightarrow$ Anger.*

However, extracting precise sentiments (in this multidimensional space) is still an unnecessary challenging task, and the majority of methods detect sentiments projected onto a single dimension of polarity, that is, Pleasantness $\Leftrightarrow$ Unpleasantness [5]. Following other methods, we record the polarity of sentiments, represented as real numbers in the range $[-1, 1]$. Negative and positive values represent negative and positive opinions respectively, while the absolute value of sentiment represents the strength of the opinion.

In order to detect contradicting opinions in general, we propose to group (aggregate) similar opinions and then measure their relative differences. Since opinions can be just any textual expressions, we can only represent their differences in a form of a distance function, by measuring on a pairwise basis the semantic distance between expressed concepts.

**Definition 3 (Opinion Distance).** *$d(O_x, O_y) = \| O_x - O_y \|$ is a function satisfying the conditions of semi-metric*

$$d(O_x, O_y) \geq 0; \quad d(O_x, O_y) = d(O_y, O_x);$$

$$d(O_x, O_y) = 0 \text{ if and only if } O_x = O_y.$$

Following the extraction of individual opinions, we can compute the aggregated opinion on some topic expressed in a collection $\mathcal{D}$ of documents (that may come from different authors, and time).

**Definition 4 (Aggregate Opinion).** *$\overline{O}$ is an opinion with the smallest sum of squared distances to other opinions within a group $\mathcal{D}$.*

**Definition 5 (Opinion Variance).** *Correspondingly, Opinion Variance $\sigma_O^2$ is the average of squared distances between opinions in $\mathcal{D}$ and $\overline{O}$*

$$\overline{O} = \operatorname*{argmin}_{O} \sum_{O_i \in \mathcal{D}} \| O - O_i \|^2 \; \sigma_O^2 = \frac{1}{n} \sum_{O_i \in \mathcal{D}} \| O_i - \overline{O} \|^2 .$$

By comparing opinion values of different collections of texts, contradictions are identified as follows:

**Definition 6 (Opinion Contradiction).** *A collection $\mathcal{D}$ of texts talking about topic $T$, is considered contradictory, if it can be partitioned into several groups of texts $\mathcal{D}_i \subset \mathcal{D}$, such that the distance between aggregate opinions of any two groups is at least $\alpha$ times greater than the maximum opinion variance*

$$\min_{i \neq j} \| \overline{O}(\mathcal{D}_i) - \overline{O}(\mathcal{D}_j) \|^2 \;\; > \;\; \alpha \cdot \max_{k} \sigma_O^2(\mathcal{D}_k). \qquad (1)$$

We define contradiction on a *pairwise* basis, where we evaluate the disagreement between two groups of documents in a collection. In this case, the similarity of concepts within each group serves as a reference point, providing a measure for the sparseness of concepts between the groups. When identifying contradictions in a document collection, it is important to also take into account the time in which these documents were published. Let $\mathcal{D}_1$ be a group of documents containing some information on topic $T$, and all documents in $\mathcal{D}_1$ were published within some time interval $t_1$. Assume that $t_1$ is followed by time interval $t_2$, and the documents published in $t_2$, $\mathcal{D}_2$, contain a conflicting piece of information on $T$. In this case, we have a special type of contradiction, called *Asynchronous Contradiction*, since $\mathcal{D}_1$ and $\mathcal{D}_2$ correspond to two different time intervals. Following the same line of thought, we say that we have a *Synchronous Contradiction* when both $\mathcal{D}_1$ and $\mathcal{D}_2$ are found in the same time interval, $t$.

**Problem 1 (Contradiction Detection).** *Partition a given collection of documents $\mathcal{D}$ into a minimal number of non-intersecting sub-groups $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$, wrt. Equation (1), and compute the level of contradiction.*

Depending on the kind of application, the above problem can be formulated for all topics in a collection, or just for a single one.

**Problem 2 (Single-Topic Contradiction Detection).** *For a given time interval $\tau$, and topic $T$, identify the time regions, where a contradiction level is exceeding some threshold $\rho$.*

**Problem 3 (All-Topics Contradiction Detection).** *For a given time interval $\tau$, identify topics $T$, which have the highest contradiction level, or the largest number of contradicting regions above some threshold.*

The time interval, $\tau$, is user-defined, whereas the length of a basic window which aggregates the documents can vary depending on the type of contradictions the application is aiming at. As we will discuss later, the threshold, $\rho$, can either be user-defined, or automatically determined in an adaptive fashion based on the data under consideration.

The latter problem is interesting if we want to consider the popularity of certain web topics. Frequent contradictions may indicate 'hot' topics, which attract the interest of the community. In this work, we focus on the solution to the first problem, since the solution to the second one is its direct extension.

Since we consider our problems from the perspective of managing sentiment information in order to enable fast query answering for aggregated sentiment analytics, the main challenge becomes in developing a contradiction detection method which is based on incrementally updatable statistical values, that can be efficiently aggregated and stored to meet the necessity of online analysis. Thus, in Section 5 we describe the instance of our problem for numeric multi-dimensional sentiments, and Section 6 outlines a scalable implementation of aggregated sentiment storage.

## 5 Contradiction Analysis

In order to be able to identify contradicting opinions we need to define a measure of contradiction. Assume that we want to look for contradictions in a shifting time window $w$. For a particular topic $T$, the set of documents $\mathcal{D}$, which we use for calculation, will be restricted to those, that were posted within the window $w$.[2] We denote this set as $\mathcal{D}(w)$, and $n$ as its cardinality, $n = |\mathcal{D}(w)|$.

Following Definition 6, we can derive a contradiction measure $C_{Ent}$ from the entropy of clustering, based on the number and sizes of contradicting groups: the largest contradiction occurs when there are many groups of equal sizes

$$C_{Ent} = -\sum_{\mathcal{D}_i \in \mathcal{D}} \frac{|\mathcal{D}_i|}{|\mathcal{D}|} \cdot \log \frac{|\mathcal{D}_i|}{|\mathcal{D}|}. \quad (2)$$

Still, the performance requirements of large-scale analytics require operating with aggregated data instead of individual clusterings. We therefore concentrate our attention on the following statistical measures of contradiction:

The *sentiment mean*, $\mu_S$, is calculated as $\mu_S = \frac{1}{n}\sum_{i=1}^{n} S_i$. It can be easily proven that $\mu_S$ has the lowest sum of squared distances to sentiments in the collection, that is, it conforms to our definition of Aggregated Sentiment. A value of $\mu_S$ close to zero implies a high level of contradiction because of positive and negative sentiments compensate each other. However, a problem with the above way of calculating the aggregated sentiment arises when there exists a large number of documents with very low sentiment values (neutral documents). In this case, the value of $\mu_S$ 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.

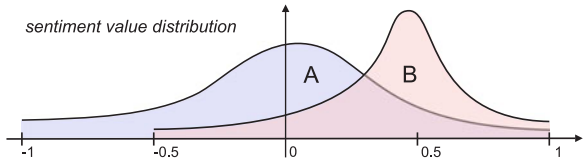


Fig. 4. Example of diverse sentiment distributions.

The *sentiment variance*, $\sigma_S^2$, is defined as the average of squared distances between sentiments and their mean: $\sigma_S^2 = \frac{1}{n}\sum_{i=1}^{n}(S_i - \mu_S)^2$. According to this definition, when there is a large uncertainty about the aggregated sentiment of a collection of documents on a particular topic, the sentiment variance is large.

The sentiment mean and variance can be expressed using first- and second-order moments of sentiment $M_1 = \sum_{i=1}^{n} S_i$ and $M_2 = \sum_{i=1}^{n}(S_i)^2$, giving us the following formulas for sentiment statistics:

$$\mu_S = M_1/n; \quad \text{and} \quad \sigma_S^2 = M_2/n - \mu_S^2. \quad (3)$$

We demonstrate the effect of outlined measures in Fig. 4, featuring two example sentiment distributions. Distribution A with $\mu_S$ close to zero and a high variance indicates a very contradictive topic. Distribution B shows a far less contradictive topic with sentiment mean $\mu_S$ in the positive range and low variance. For example, a group of documents with $\mu_S$ close to zero and a high variance (distribution A on the Fig. 4) will be very contradictive, and another group with sentiment $\mu_S$ shifted to negative or positive with low variance is likely to be far less contradictive (distribution B on the Fig. 4). When assuming a large number of neutral sentiments in the collection, we have two opposite trends: the average sentiment moves towards zero and sentiment variance decreases. If these trends will compensate each other, the neutral documents would not affect the contradiction value much.

Evidently, we need to combine mean and variance of sentiments (expressed in the same units) in a single formula for computing the contradiction value $C$

$$C = \sigma_S^2/\mu_S^2. \quad (4)$$

The above formula captures the intuition that contradiction values should be higher for distributions where the aggregated sentiment value is close to zero, and sentiment variance is large. This property follows from the criteria for opinion contradictions (Formula 1), as demonstrated by Lemma 1 below. Moreover, we can compute opinion contradictions (Formula 2) on aggregated sentiment statistics using their equivalent representation by sentiment classes according to Lemma 2. Using this lemma, we can evaluate both measures on the same data.

2. This work uses windows of days, weeks, months, and years.

**Property 1.** *Assuming that the two collections of sentiments, $A$ and $B$, demonstrated in Fig. 4, are the components of a bimodal sentiment distribution, and that $A$ and $B$ have $n_a$ and $n_b$ sentiments in each, distributed with parameters $(\mu_a, \sigma_a)$ and $(\mu_b, \sigma_b)$, we can calculate the mean value $\mu_S$ and the variance $\sigma_S^2$ of their aggregated sentiment distribution as follows:*

$$n = n_a + n_b,\ M_1 = M_1^a + M_1^b \text{ and } M_2 = M_2^a + M_2^b$$
$$\mu_S = \frac{n_a\mu_a + n_b\mu_b}{n_a + n_b};\ \sigma_S^2 = \frac{n_a\sigma_a^2 + n_b\sigma_b^2}{n_a + n_b} + \frac{n_a n_b(\mu_a - \mu_b)^2}{(n_a + n_b)^2}.$$

**Lemma 1.** *If a bimodal sentiment distribution satisfies Definition 6, then its contradiction level C is not lesser than the relative separation of the components.*

**Proof.** Using the formulas of the aggregated values $\mu_S$ and $\sigma_S^2$ in our measure of contradiction, and discarding the variance component from the nominator according to Definition 6, we obtain the following expression, similar to sentiment polarisation in [18]

$$C = \frac{\sigma_S^2}{\mu_S^2} = \frac{(n_a + n_b)(n_a\sigma_a^2 + n_b\sigma_b^2) + n_a n_b(\mu_a - \mu_b)^2}{(n_a\mu_a + n_b\mu_b)^2} > \frac{n_a n_b(\mu_a - \mu_b)^2}{(n_a\mu_a + n_b\mu_b)^2}, \quad (5)$$

where the difference $\mu_a$-$\mu_b$ is normalized by the harmonic sentiment average. □

Now it can be clearly seen that a larger separation between sentiment distributions results in a higher contradiction value. Taking into account the limited range of sentiment values, this distance is the largest when sentiment means are of the opposite polarities. In this case, the two sentiment distributions compensate each other and the denominator becomes very small, obtaining $C \gg 1$.

We can also represent a sufficiently large sentiment distribution as a mixture of the three groups of sentiments—positive, negative and neutral, and apply the opinion contradiction formula on them. Lemma 2 allows us to compute the sizes of these groups $(n_p, n_n, n_0)$ -as shown in the proof- from the statistical moments of sentiment, as follows:

**Lemma 2.** *If a collection of sentiments with statistical moments $M_1$ and $M_2$ has a size $n \gg 1$, it can be equivalently represented by another collection, with the same size and statistical moments, containing $n_p$ positive and $n_n$ negative sentiments (with absolute values $\alpha$) and $n_0$ neutral sentiments (zero values).*

**Proof.** Let $|S_i| = \alpha$, $n = n_p + n_n + n_0$, $M_1 = \alpha(n_p - n_n)$, $M_2 = \alpha^2(n_p + n_n)$; then $2n_p = (M_2 + \alpha M_1)/\alpha^2, 2n_n = (M_2 - \alpha M_1)/\alpha^2$, $n_0 = n - M_2/\alpha^2$; where $\sqrt{M_2/n} \le \alpha \le M_2/|M_1|$, and $\sqrt{M_2/n} \le M_2/|M_1|$ since $\sigma_S^2 \ge 0$; Sentiment value $\alpha = \text{argmax}(C)$. □

In the above representation, larger difference between positive and negative sentiments (larger values of $\alpha$) leads to smaller numbers of $n_p$ and $n_n$, and increased $n_0$. Therefore, it is necessary to pick the value of $\alpha$, which results in *maximum entropy* distribution of sentiments between classes, that is, in the maximum opinion contradiction (Formula (2) is based on entropy).

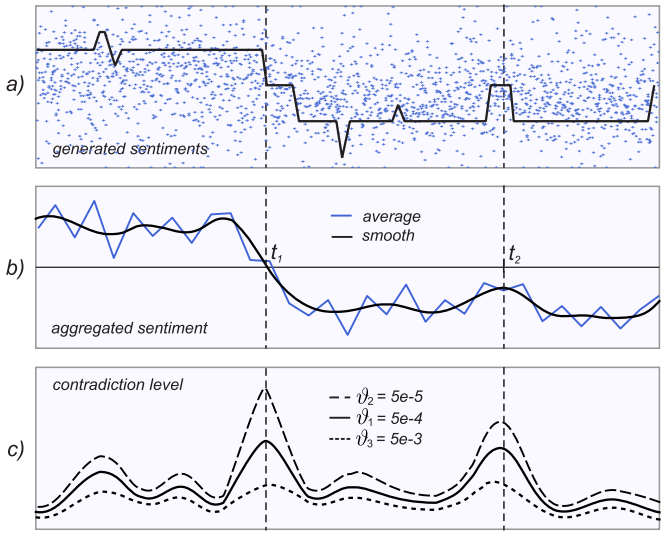



Fig. 5. Sentiment data with artificial contradictions.

### 5.1 Computing Contradictions

So far we demonstrated that sentiment contradiction (Formula (4)) corresponds to our definitions and can be computed directly from aggregated sentiment polarity values. Nevertheless, this formula should be extended for being suitable to real application scenarios, where extracted sentiments usually contain noise and their flow over time is irregular. We observe that this formula produces unbounded contradiction values (i.e., they can grow arbitrarily high as $\mu_S$ approaches zero), and that it also does not account for the number of sentiments $n$, that is, for the significance of sentiment statistics. For instance, in the extreme case when $\mathcal{D}(w)$ contains only two documents with opposite values, $C$ will become infinitely high, and thus incomparable to the contradiction value of any other set of documents with higher cardinality. While the first problem (of the infinite contradiction scale) can be addressed with the help of a regularizing constant added to the denominator, the second problem (of statistical significance) is important for small-scale applications or for streams of sentiments with the irregular flow. We propose to cope with this problem by accounting on the significance of statistics involved in the calculation of $C$ using the logistic weight function $W(n)$ [8]

$$C = \frac{\vartheta \cdot \sigma_S^2}{\vartheta + \mu_S^2} W(n). \quad (6)$$

In the denominator, we add a small value, $\vartheta \neq 0$, which limits the level of contradiction when $\mu_S$ is close to zero. The same value $\vartheta$ also scales the nominator to ensure that contradiction values are always contained within the interval $[0; 1]$. Fig. 5c shows that $\vartheta$ has a kind of "local contrast" effect on contradiction values. Smaller $\vartheta$ values emphasize contradiction points with $\mu_S$ close to zero, for example changes of opinion. Larger $\vartheta$ values mask this difference, making levels of contradictions more equal. In this study, we used a value of $\vartheta = 5 \times 10^{-4}$, which was effective for its purpose, exhibiting a stable behavior across datasets, without distorting the final results.

$W(n)$ is weight function that provides a multiplicative factor in the range $[0; 1]$, indicating the significance of a contradiction (Fig. 6 plots $W$ as a function of $n$):

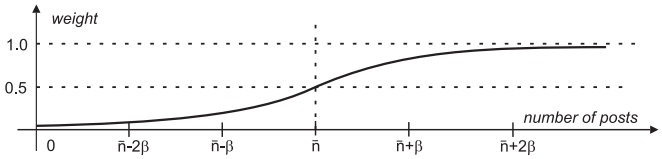



Fig. 6. The weight function used in the criteria.

$$W = \left(1 + exp\left(\frac{\overline{n} - n}{\beta}\right)\right)^{-1}$$

Here the constant $\overline{n}$ reflects the expected number of documents in the window, and $\beta$ is a scaling factor. Using $W$ we can effectively limit $C$ when there is a small number of documents, and weigh it more when that number is large.

While $W(n)$ addresses the problem of significance of contradiction values, there exists another source of error, which can be attributed to the irregularity of aggregated sentiment. This irregularity can be explained by considering that population samples that contribute to aggregated sentiments are quite different across adjacent time intervals. Indeed, people tend to publish at a particular rate, and the likelihood that they will re-state their sentiment shortly after the first publication is low for small aggregation windows. On one hand, increasing the window size at the same granularity level can help reducing such noise, but at the same time it will decrease the resolution of our analysis, allowing to identify only long-lasting contradictions. On the other hand, applying a sliding window of a larger size on a smaller granularity requires substantially more resources for storage and computation. To cope with this problem, we propose to use *local regression smoothing* [25], which computes a smooth regression trend with regard to sentiment observations and their variance. The regression trend ensures the continuity of sentiment values, but unlike sliding window based smoothing, it preserves the sharpness of significant sentiment deviations (by considering sentiment variance). In particular, we apply the cubic polynomial spline regression from SSJ library[3] with regression parameter of 0.5 and inverted variance for weights. Nevertheless, other smoothing methods and their parameters can be applied depending on the nature of data and processing requirements.

Fig. 5 shows the operation of the proposed contradiction function. To better illustrate this, we use one of the time series from our synthetically generated dataset (described in Section 7). The graph at the top (Fig. 5a) shows generated sentiments. The bold line in this graph depicts the custom trend, showing an initial positive sentiment that later changes to negative (at time instance $t_1$), which represents an asynchronous contradiction (change of sentiment) that manifests itself across the entire dataset. There is also a point around time instance $t_2$, where the sentiments are divided between positive and negative, a situation representing a synchronous contradiction. As can be seen in Fig. 5b, a smoothed trend of $\mu_S$ (using regression smoothing) captures the aggregated sentiment better than the simple average, effectively reducing noisy fluctuations. The graph in Fig. 5c shows the contradiction value obtained using smoothed mean and variance of sentiments. In this

3. http://www.iro.umontreal.ca/~simardr/ssj/indexe.html

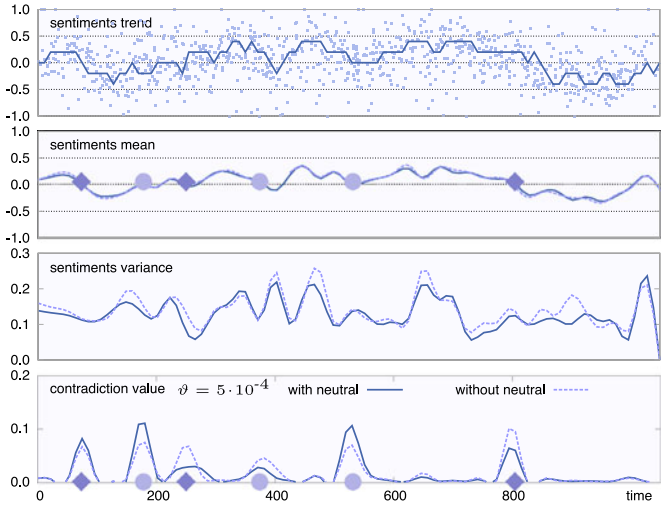



Fig. 7. The effect of neutral sentiments for detecting synchronous (●) and asynchronous (◆) contradictions.

case, $C$ correctly identifies the two contradictions at points $t_1$ and $t_2$, where the values of $C$ are the largest. In this case, using simple aggregated values of sentiments $\mu_S$ straight away can result in $C$ reporting noisy fluctuations of sentiments as contradictory.

Subjective sentences take a considerably small part in the text when compared to objective statements. So neutral sentiments usually shift the aggregate sentiment towards zero, masking contradictions. Our contradiction formula is designed to compensate such effects by exploiting the sentiment variance. We demonstrate such behavior on another synthetic dataset shown in Fig. 7. The bottom graph shows that the proposed formula can successfully identify the main contradicting regions, marked by dots, both with or without neutral sentiments. Nevertheless, in their perception of contradiction, people usually account for the relative amount of neutral statements. Hence, they do not consider as contradictory regions containing mostly neutral sentiments (as we observed in Section 1). This should be taken into account if subjectivity filtering is applied upon sentiment extraction, removing neutral sentiments from the distribution. In such cases, it is possible to tune the sensitivity of our contradiction measure by setting the parameter $\vartheta$.

### 5.2 Detecting Contradictions

Since weighting on the number of sentiments addresses the problem of statistical significance and local regression smoothing helps dealing with irregular data, the problem of detecting contradictions using our formula boils down to picking the right sentiment aggregation window size and contradiction threshold value.

In the case of synchronous contradictions, when the community at every particular interval in time has different opinions about the same topic, contradictions can be determined easily with any suitable time window. However, sometimes the community has a solid opinion in one time period, and later changes it, so in another time period it has the opposite opinion, resulting in an asynchronous contradiction. This type of contradiction can only be discovered using a time window large enough to gather posts from the two different periods. Moreover, if for some shifts of

opinion there exists a gap in time between positive and negative posts, the detection becomes highly dependent on the time window and on the order in which posts were published. By using a small time window we will likely get only a small peak of contradiction at the moment when the community has changed its opinion, because the transition between opposite opinions is slow enough to result in any significant difference of opinions at any particular time interval. Thus, the hierarchical refinement of time intervals from large to small is particularly important for the discovery of asynchronous contradictions.

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, $\rho$, 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 $\varrho_w$ for a topic $T$ in time window $w$ is based on the contradiction value $C_{w_p}$ that has been calculated for $T$ in the parent time window of $w$, $w_p$, and is defined as $\varrho_w = p \cdot C_{w_p}$. In our experience with real datasets, $p$ values between 0.5-0.7 work well. In this work, we use $p = 0.6$.

Adaptive threshold helps to detect interesting contradictions that occur in different time granularities and across topics, even if these contradictions do not have the largest values overall. This is particularly important when a single, fixed threshold value cannot detect all contradictions across time, or when the user is unsure about which threshold to choose. 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 the adaptive threshold is changing as we navigate the timeline, and it provides even discrimination of peaks of contradiction both in high- and low-contradicting regions. Moreover, it 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.

## 6 Diverse Sentiment Aggregation

So far we have described a technique to identify contradictions from sentiment aggregates. But our final goal is to process large-scale streams of sentiments, what requires scalable methods. To this end, we demonstrated the need to analyze sentiment information on each topic across different time windows. Assuming this requirement, scalability may be achieved by storing pre-computed values for windows of different size.

We now turn our attention to the problem of organizing all these data in 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 both our contradiction formulas use additive statistics, which can be easily updated and aggregated. This property of the contradiction formulas 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.

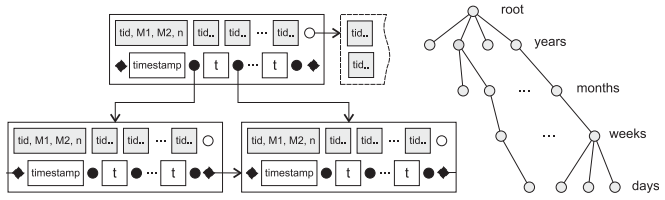



Fig. 8. Physical and hierarchical structure of CTree nodes.

Formula (2) is based on group counts, while Formula (6) is based on mean and variance, which can be computed from the first- and second-order moments of sentiments, as shown in Formula (3). Based on this representation, we can rewrite Formula (6) using the sentiment moments $M_1$ and $M_2$, as follows:

$$C = \frac{\vartheta(nM_2 - M_1^2)}{\vartheta n^2 + M_1^2} W(n). \tag{7}$$

The need to analyze contradictions at different time granularities calls for a hierarchical structure for contradiction storage, like the one illustrated in Fig. 8 (right). 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 contradiction values from aggregated data.

One possible solution is to use the above time-tree structure for each topic separately. It allows to achieve scalability on the number of topics, and has a good performance when looking for contradictions within a single topic. However, it involves high update costs, because for each text the data structure needs to be parsed as many times as there are topics in that text. In addition, it renders all-topic queries ineffective, because for each topic we need to navigate through a time structure in order to find the right interval. An alternative solution would be to store contradiction values for different topics under the same time-tree structure. This solution does not suffer from the disadvantages mentioned earlier, and is the solution of choice for this study.

We now introduce the Contradiction Tree (CTree) for managing the information on sentiments and contradictions. The CTree is organized around the sentiment moments, $M_1$ and $M_2$, and a hierarchical segmentation of time, as outlined in Fig. 8. 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 documents, whose timestamp is contained in this time window. The internal structure of the CTree nodes is illustrated in Fig. 8 (left). As the figure shows, a CTree node stores the following information: (a) for each topic, the topic id, $tid$, the number of documents, $n$, 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 Fig. 8 (left)). 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.

Algorithm 1 outlines the algorithm that uses the adaptive threshold to retrieve contradictions. It needs a single pass over the collection of pages of the specified granularity, $l$, that fall inside the time interval, $\tau$ of the query. Note that contradiction values are computed from the information stored in the node using Formula (7). The type of contradiction is identified by comparing signs of sentiments for adjacent nodes. In our implementation, we additionally do not visit child nodes whose parents are not contradictory (we omit this detail from the algorithm for ease of presentation).

**Algorithm 1.** CTree Access

**Input**: Topic $T$, Time interval $\tau$, Granularity $l$
**Output**: List of contradictions $\{(time, contradiction, type)\}$
Set output contradictions $\mathcal{C} = \emptyset$;
Navigate nodes at parent gran:
**forall** $r \in \tau$, $r.gran = l - 1$ **do**
  **forall** *nodes* $r_i \in r.childNodes$ **do**
    **if** $r_i \in \tau$ and $r_i.C^T > p \times r.C^T$ **then**
      **if** $r_{i-1}.S^T \times r_i.S^T \leq 0$ **then**
        $type$ = "async";
      **else**
        $type$ = "sync";
      **end**
      $\mathcal{C} = \mathcal{C} \cup (r_i, r_i.C^T, type)$;
    **end**
  **end**
**end**
Arrange $\mathcal{C}$ by contradiction count or level;
**return** $\mathcal{C}$;

The sentiment statistical moments $\{n, M_1, M_2\}$ allow us to incrementally maintain the CTree in the presence of updates. In order to reduce update costs, we propose first to accumulate several updates and then submit them in a batch, as shown in Algorithm 2. When new documents arrive, they are aggregated in time windows of the finest granularity of the CTree. Then, these aggregated values are used to update the counts and topic sentiment moments of all CTree nodes containing respective time windows. The update cost for each batch of aggregated documents depends on the depth of the CTree, $d$, the number of updated nodes, and the topic position in disk pages. In the worst case it matches $O(d\frac{|T|}{h})$, where $h$ is the maximum number of topics stored in a single node.

**Algorithm 2.** CTree Update

**Input**: Topic sentiments series $\{t_i, T_i, S_i^T\}$, interval $\tau$
**define** `update` as ($\tau$, $n$, $M_1$, $M_2$);
**define** `updateset` as a set {`update`};
Aggregate sentiments over smallest $\tau$:
$\mathcal{S}_\tau^T = \{S_i^T \mid t_i \in \tau, \tau.gran = 0\}$
$upd$ = {($\tau$, $n^T$=$|S_\tau^T|$, $M_1^T$=$\sum S_\tau^T$, $M_2^T$=$\sum S_\tau^{2T}$)};
**call** `UpdateNode` ($rootNode, upd$);

**function** `UpdateNode`(`node` $r$, `updateset` $upd$);
**if** $r.childNodes \neq \emptyset$ **then**
  Set $updResult$ = ($upd.\tau$, 0, 0, 0);
  **forall** *node* $r_i \in r.childNodes$ **do**
    Set `updateset` $upd_i = \emptyset$;
    **forall** `update` $u \in upd$ **do**
      **if** $u.\tau \in r_i$ **then** $upd_i$ += $u$;
    **end**
    $updResult$ += `UpdateNode`($r_i$, $upd_i$);
  **end**
**else** $updResult = \sum_{i=1}^{|upd|} upd_i$;
r.($n^T$, $M_1^T$, $M_2^T$) += $updResult^T$;
**return** $updResult$;

## 7 Experimental Evaluation

In this section, we report the results of our experimental evaluation on synthetic and real datasets. The objectives of the experiments we conducted were to: analyze the quality of the approach; study its usefulness from a user perspective; and finally, study the scalability of our solution.

The performance evaluation was conducted on a desktop computer with a dual core CPU. Our algorithms were implemented in Java 1.6.13. The database we used for the baseline was IBM DB2 9.5.2.[4]

Specifically for the evaluation of accuracy and performance of our method, we generated a synthetic dataset containing time series of sentiments following the artificial trend with opinion shifts, contradictions and a controlled amount of noise. To create this dataset we generated a large volume of sentiments with time stamps following the Poisson distribution with the average rate from 1 to 10 sentiments per day, and with polarities sampled using normal distributions. We have chosen these distributions because they are simple, and still resemble the real data. A particular fraction of generated sentiments followed a planted trend with dispersion 0.125, while the rest, controlled by the noise parameter, were distributed randomly with dispersion 0.5 and mean 0.0. The relative amount of noise sentiments varied from 0 to 40 percent with a step of 10 percent. We generated 1,000 sentiment trends, and stored the corresponding original time series (with 0 percent of random noise) in the CTree, also

4. Our CTree implementation and datasets can be found at http://www.mi.parisdescartes.fr/~themisp/ctree/

TABLE 1
Contradiction Detection Performance

| Method | Sent-C | Ent-C2 (Ent-C3) | SVM (SVM hist) | LR (LR hist) | Baseline |
|---|---|---|---|---|---|
| Accuracy | **82.0** | 79.7 (70.3) | 78.9 (79.7) | 68.8 (66.4) | 50.0 |
| Precision | **93.6** | 85.2 (67.6) | 91.1 (93.2) | 72.2 (66.2) | 50.0 |
| Recall | 68.8 | 71.9 (**78.1**) | 64.1 (64.1) | 60.9 (67.2) | 100.0 |
| F-Measure | **79.3** | 78.0 (72.5) | 75.2 (75.9) | 66.1 (66.7) | 66.7 |

duplicating them and adding noise for each of the above parameters. Overall, we stored 5,000 time series in the CTree.

In our evaluation we also use two real datasets. The first contains user comments from Drug ratings, YouTube, and Slashdot, which were manually annotated for sentiment; we have described this dataset in detail in the Introduction. The second dataset comes from Twitter, containing approximately 7 million tweets on 30 trending topics; more details are provided in Section 7.4.

### 7.1 Contradiction Detection Accuracy

We evaluate the accuracy of our measures for contradiction detection on the real manually labelled dataset of sentiment distributions [8] described in Section 1, comparing them to several machine learning classifiers from Weka data mining tool. We used the same dataset both for training and testing, reporting the average statistics for 10-fold cross-validation, where 90 percent of data were used for training and 10 percent for testing on every split iteration.

As the main alternative to our methods we chose an SVM classifier (nu-SVC type using radial kernel), with its parameters optimized for the best precision. In addition to SVM, we used the Logistic Regression (LR) classifier. Both classifiers used feature vectors based either on means and variances of sentiment distributions (the default) or on complete histograms (reported as *hist*). In contrast, our classifiers based on the sentiment contradiction formula (Sent-C) and the entropy contradiction formula (Ent-C) use only statistical aggregates as input data, which are much less descriptive than histograms. Similarly to machine learning classifiers, our methods used thresholds yielding the best accuracy on training data.

The results of our evaluation are shown in Table 1, where we report the overall accuracy (of instances correctly classified as contradictory or non-contradictory) and precision with recall (of instances correctly classified as contradictory). Since our dataset is balanced, the baseline accuracy and F-Measure for classifying contradictions are 50 and 66.7 percent, respectively.

The best results were achieved by our proposed approach, Sent-C, which is 3 percent more accurate than SVM. SVM-hist, which uses much more information is still performing below Sent-C, and the same is true for Ent-C2 (entropy contradiction formula with two clusters: positive and negative). Ent-C3 (entropy contradiction formula with three clusters, where we additionally take into account the neutral sentiments) resulted in better recall, but considerable precision loss, since many non-contradictory distributions were reported as contradictory. The LR method demonstrated significantly worse results and was not able to benefit from using histogram data either, most likely because it was not possible to separate the different classes in a linear space.

We should note, that in this experiment SVM methods were rather good at classifying contradictions mainly because of the cross-validation and exhaustiveness of the evaluation dataset. While the first circumstance alone required training on the 90 percent of whole data, in combination with the second one it made most of the testing samples similar to those used for training. This makes the reported precision values for these methods reading as an optimistic estimate, rather than the actual performance. On the other hand, the optimal value of the contradiction threshold used in these experiments was most often equal to 0.5 of the average contradiction level across all tested samples, indicating that our approaches are very effective even with the default setting of the adaptive threshold.

We further note that our model uses only statistical moments (Sent-C) or polarity counts (Ent-C) as input data, which are less descriptive than histograms. Furthermore, our methods in this case did not apply the significance-based weighting, since we used annotated data based on text collections of the same size. Because SVM utilizes normalized values, it cannot automatically handle situations when statistical values are not significant due to a small number of sentiments. Even when the number of sentiments is added as an additional feature, pruning on this feature can be either uncontrollably biased by other features, or very strict, depending on training data.

Even though SVM methods have demonstrated good performance, they fail at delivering several other important properties that are relevant for our problem. These include measuring the level of contradiction, filtering and ranking the result set, and automatically handling situations when statistical values are not significant (due to a small number of sentiments). Moreover, SVM methods are not able to adapt to different kinds of sentiment biases in real datasets without training. In contrast, we can control sensitivity of our method using $\vartheta$ parameter, and additionally it is possible to compensate biased sentiments by adjusting $\mu_S$, without modifying the actual values stored in the CTree.

### 7.2 Contradiction Detection on Noisy Data

We evaluate the accuracy of our method by measuring precision and recall of extracting contradictions for varying classifier performance. We randomly picked 10 original time series from our synthetic dataset, each having five noise versions, with the ratio of random sentiments ranging from 0 to 40 percent, roughly equivalent to a sentiment classifier precision varying from 95 to 55 percent.[5] We note that for large-scale applications, high sentiment classification precision is more important than high recall, since the task is to accurately measure the average sentiment of a sample population. Contradiction precision is computed as the percentage of the extracted contradiction intervals that match to the correct ones. Contradiction recall is computed as the percentage of the true contradiction intervals, which were actually extracted. In Fig. 9 we demonstrate the overall

5. Since some sentiment variation is present in the synthetic time series even at zero noise setting, and since the equivalent precision depends on the distance between trend and mean noise sentiments.

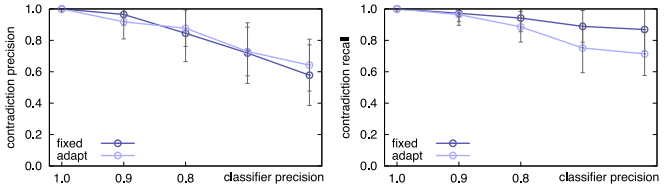



Fig. 9. Precision and recall of contradiction detection versus sentiment classifier performance.

accuracy of our method. We used a constant threshold for all time series (0.05 fixed and 2.5 adaptive) and compared the extracted contradictions for noisy time series with the ones for noise-free time series. We observe that the precision is gradually decreasing with noise, yet maintaining a usable level until about 20 percent of random noise, which is feasible for state-of-the-art sentiment classifiers. The recall for a fixed threshold remains very high for all settings, while the recall for adaptive threshold drops slightly due to the increased absolute threshold value.

In order to evaluate the accuracy of asynchronous contradictions, we manually identified major changes of opinion in the selected time series (without noise), and used them as the ground truth. We did not consider as asynchronous contradictions such changes of opinion, where the sentiment time series shortly crosses the zero line and then goes back (e.g., such as points 2 and 5 on Fig. 7). For these experiments we used a fixed 0.05 contradiction threshold, and a 0.5 coefficient for time series smoothing. Applying an adaptive threshold instead of a constant threshold in this case does not yield a dramatic improvement of precision of asynchronous contradictions, since their level only depends on variance (sentiment mean is zero at the change point), which remains almost constant for large aggregation granularities.

The graph in Fig. 10 shows the accuracy of detecting asynchronous contradictions with and without regression smoothing applied. It also demonstrates the performance of using single (10 days) and multiple (10 days, 30 days) aggregation granularities. The latter method detects contradictions at larger granularity and combines them with those detected at smaller granularity, in order to efficiently capture both rapid (local) and slow (global) changes of sentiments.

We observe that both methods correctly identify a large fraction of the contradictions at all noise settings. The recall is varying from above of 90 to about 65 percent, and stays firmly above 75 percent for the smoothing multi-granularity version at low to mid noise levels, meaning that our method is applicable to and useful for information retrieval purposes. The fact that recall values never reach 100 percent in this particular experiment reveals that a small fraction of opinion changes can not be detected even at the granularity of 30 days. Precision values are significantly better for the smoothing multi-granularity version of our method (95 percent) compared to either average or single-granularity versions (50-75 percent).

The obtained results clearly indicate the superiority of the multi-granularity time series analysis method over the single-granularity method. In its turn, the multi-granularity method further justifies the necessity to use our hierarchical storage for sentiments, CTree, which provides fast simultaneous access to different sentiment aggregation levels.

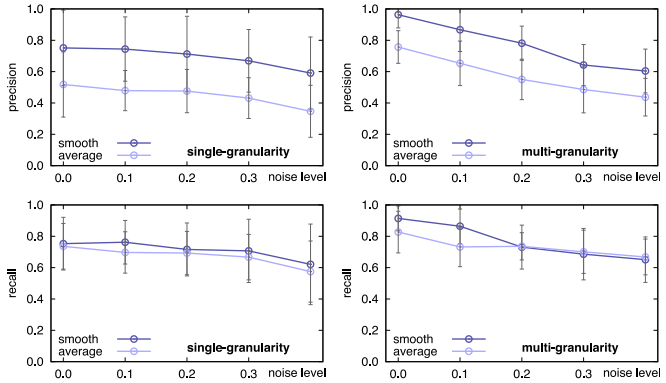



Fig. 10. Precision and recall of asynchronous contradictions versus single or multiple aggregation granularities, and smooth versus average sentiment.

### 7.3 Evaluation of Scalability

We evaluate the scalability of CTree for the topic contradiction Problem (2), where we want to identify the contradictions of a *single topic* within some time interval, and for the all topic contradictions Problem (3), where we are interested in doing the same for *all topics*. The obtained results, however, are indicative for other similar types of queries to our storage, which require *ad-hoc* navigation to time intervals, either without their parents (equiv. to fixed-threshold contradiction queries), or with the simultaneous access to parent-level statistical aggregates (equiv. to adaptive-threshold contradiction queries).

During this study, the parameters of the contradiction formula were at their default values as described in Section 5. Changing formula's parameters will enlarge or reduce the number of contradictions being detected, but the computational efficiency will be the same. Performance of our approach does not depend on the value of threshold because we are not storing pre-computed contradiction values, and so the database is unable to apply indices or filtering on this parameter. Fixed and adaptive threshold approaches, however, return slightly different sets of contradictions. The first one returns largest contradictions themselves, and the second returns contradictions that are greater than $p$-times values of their respective parent intervals. The value of $p$ was empirically set at 0.6 to return a result set with an average size equal to the one when using a fixed threshold. This allows us to compare the relative performance of both methods.

To test the scalability, we generated sets of 25 queries for Problems (2) and (3), using granularities and topic ids drawn uniformly at random. We compare our solution, CTree, against a database implementation, Cdb, which stores the contents of CTree nodes in a single database table. For this table, we created the appropriate database indices (for time, granularity and topics), based on the performance profiling suggested for our queries.

In the first set of experiments, we measured the time needed to execute single- and all-topic queries as a function of the time interval, $\tau$, and the granularity of the time windows (Fig. 11). 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

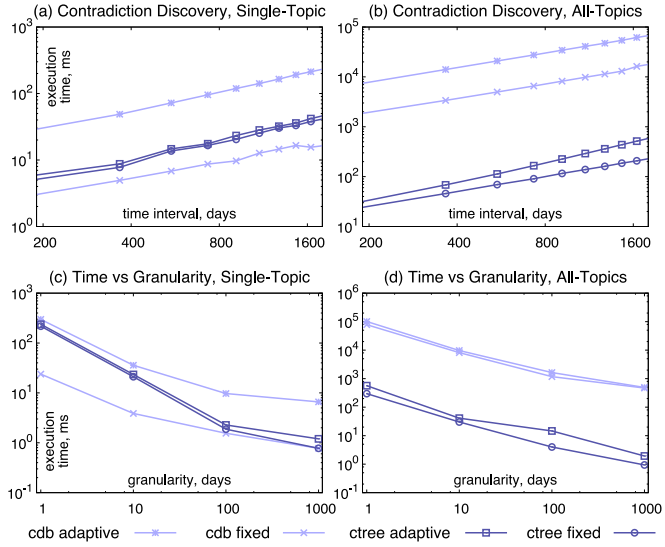



Fig. 11. Single-topic versus all-topics queries scalability.

pronounced for Cdb, because it involves an extra join for obtaining the parent time window. On the other hand, the same result 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 Figs. 11a and 11b) scale linearly with the size of $\tau$. 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 $\tau$. For single-topic queries with fixed threshold, the database is able to use all its indices (i.e., on topic, time windows, and granularity) to answer the queries, therefore, achieving fast response times. In all other cases (i.e., all-topic queries, or adaptive threshold), the CTree approach performs up to three orders of magnitude faster than Cdb. This pronounced difference is explained by the ability of CTree to access sequential time intervals without having to navigate through the index for each one of them—a situation taking place in the case of Cdb.

Figs. 11c and 11d 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 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 Cdb as with the previous experiments.

Finally, we measured the time needed to update the CTree and Cdb. In Fig. 12, 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 update cost scales linearly with the number of topics in the system and that CTree performs four times faster than Cdb. This agrees well with the discussion in Section 6, that access cost is proportional to the number of topics.

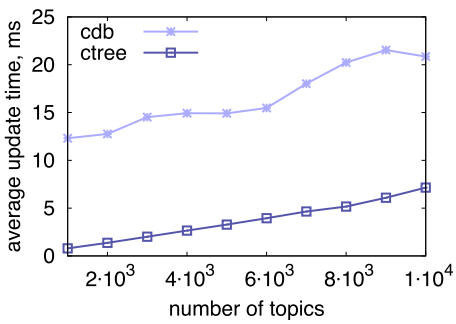



Fig. 12. Update time versus number of topics stored.

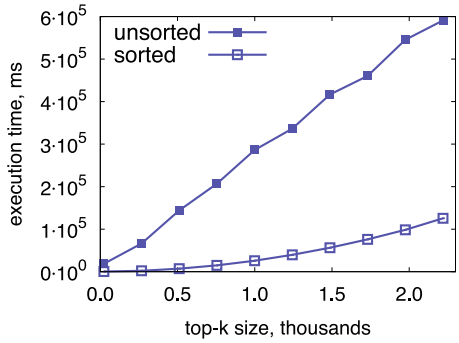



Fig. 13. Top-k queries execution time.

Since the database solution stores information about all the topics in the same table and treats them uniformly, its performance can not be improved for the cases where some topics are more popular (receive more queries) than others. Therefore, the uniform distribution of topic ids used in our experiments favors the database approach. In contrast, CTree can arrange topics using different orderings (e.g., sorted by popularity or contradiction level), and do so independently for each time interval.

To have a notion on how significantly the performance of the CTree at answering top-k queries improves when topics are stored prearranged by their level of contradiction, we performed an experiment on a range of "all-topics" queries with random parameters. In Fig. 13, we plot the average execution times for Problem (3) using a varying limit on the number of returned contradictions. It is clearly visible that a sorted version of the CTree performs on average 6 times faster than the original one. However, this approach reduces performance for the ad-hoc topic access in the case when topics are arranged by contradictions rather than popularity.

### 7.4 Evaluation on Twitter

Finally, to demonstrate the usefulness of our approach, we selected 30 trending topics from Twitter, which featured the most prominent events for the period of half a year, from June 2009 till December 2009. This dataset contained approximately 7 million tweets, which we assigned with sentiment labels by SentiStrength, as more appropriate for short messages, and applied CTree algorithm on them to detect contradictions. We then used a 1-day aggregation for the time series of sentiments and automatically labeled the highest contradictions by TF-IDF keywords. The result of our processing can be seen in Fig. 14, demonstrating sentiments about the Large Hardon Collider, at the time when it was malfunctioning. In this example, we see that people started to talk negatively in the aftermath of the first experiments (marked *"collision"*), while the news about the record beam energy (marked *"record energy"*) pushed sentiments back to neutral,

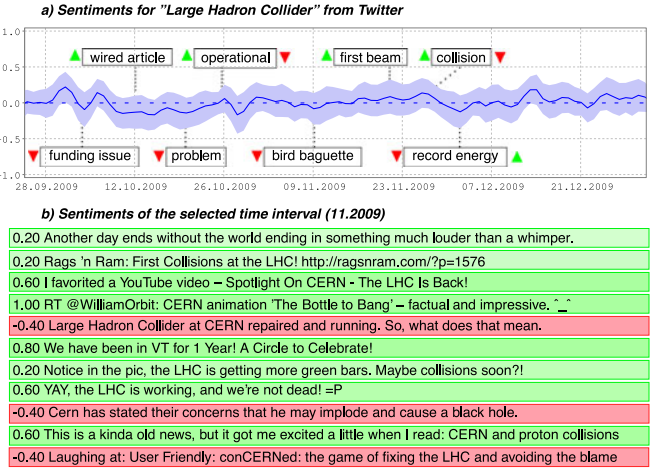



Fig. 14. Example of detected sentiment changes.

showing that our approach is able to produce meaningful output on such large-scale noisy data, as tweets.

## 8 Conclusions

In this paper, we formally define the concepts of sentiment and opinion contradictions and the problems of their detection with respect to the time dimension for a single or all topics. We propose approaches for information-preserving storage of diverse sentiments and for detecting contradictions for large-scale and noisy data sources, which is the first general and systematic solution to the problem. An experimental evaluation with synthetic and real data demonstrates the applicability, usefulness, and efficiency of the proposed solution.

## References

[1] B. Liu, *Sentiment Analysis and Opinion Mining*. San Rafael, CA, USA: Morgan & Claypool Publishers, 2012.
[2] M. Tsytsarau, S. Amer-Yahia, and T. Palpanas, "Efficient sentiment correlation for large-scale demographics," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2013, pp. 253–264.
[3] M. Tsytsarau, T. Palpanas, and M. Castellanos, "Dynamics of news events and social media reaction," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 901–910.
[4] E. Cambria, H. Wang, and B. White, "Guest editorial: Big social data analysis," *Knowl.-Based Syst.*, vol. 69, pp. 1–2, 2014.
[5] M. Tsytsarau and T. Palpanas, "Survey on mining subjective data on the web," *Data Mining Knowl. Discovery*, vol. 24, pp. 478–514, 2012.
[6] K. Lerman, S. Blair-Goldensohn, and R. Mcdonald, "Sentiment summarization: Evaluating and learning user preferences," in *Proc. 12th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2009, pp. 514–522.
[7] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 813–830, Mar. 2016.
[8] M. Tsytsarau, T. Palpanas, and K. Denecke, "Scalable detection of sentiment-based contradictions," in *Proc. 1st Int. Workshop Knowl. Diversity Web*, 2011, pp. 9–16.
[9] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intell. Syst.*, vol. 28, no. 2, pp. 15–21, Mar. 2013.
[10] R. Xia, F. Xu, C. Zong, Q. Li, Y. Qi, and T. Li, "Dual sentiment analysis: Considering two sides of one review," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 8, pp. 2120–2133, Aug. 2015.
[11] K. Denecke and M. Brosowski, "Topic detection in noisy data source," in *Proc. 5th Int. Conf. Digit. Inf. Manage.*, 2010, pp. 50–55.
[12] J. Zeng, Z.-Q. Liu, and X.-Q. Cao, "Fast online EM for big topic modeling," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 675–688, Mar. 2016.
[13] S. Tan, et al., "Interpreting the public sentiment variations on Twitter," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1158–1170, May 2014.
[14] S. Liu, X. Cheng, F. Li, and F. Li, "TASC: Topic-adaptive sentiment classification on dynamic tweets," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 6, pp. 1696–1709, Jun. 2015.
[15] M. C. de Marneffe, A. N. Rafferty, and C. D. Manning, "Finding contradictions in text," in *Proc. Annu. Meeting Assoc. Comput. Linguistics Human Language Technol. Conf.*, Jun. 2008, pp. 1039–1047.
[16] A. M. Popescu and M. Pennacchiotti, "Detecting controversial events from Twitter," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, 2010, pp. 1873–1876.
[17] M. Thelwall, K. Buckley, and G. Paltoglou, "Sentiment in Twitter events," *J. American Soc. Inf. Sci. Technol.*, vol. 62, no. 2, pp. 406–418, 2011.
[18] A. Morales, J. Borondo, J. Losada, and R. Benito, "Measuring political polarization: Twitter shows the two sides of venezuela," *Chaos*, vol. 25, no. 3, pp. 1–11, 2015.
[19] M. Dînşoreanu and R. Potolea, "A scalable approach for contradiction detection driven by opinion mining," in *Proc. Int. Conf. Inf. Integr. Web-Based Appl. Services*, 2013, pp. 7–15.
[20] M. D. Choudhury, H. Sundaram, A. John, and D. D. Seligmann, "Multi-scale characterization of social network dynamics in the blogosphere," in *Proc. 17th ACM Conf. Inf. Knowl. Manage.*, 2008, pp. 1515–1516.
[21] I. Varlamis, V. Vassalos, and A. Palaios, "Monitoring the evolution of interests in the blogosphere," in *Proc. 24th Int. Conf. Data Eng. Workshops*, 2008, pp. 513–518.
[22] M. Tsytsarau and T. Palpanas, "NIA: System for news impact analytics," in *Proc. KDD Workshop Interactive Data Exploration Analytics*, 2014, pp. 127–129.
[23] R. Johansson and A. Moschitti, "Extracting opinion expressions and their polarities: Exploration of pipelines and joint models," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics Human Language Technologies*, 2011, pp. 101–106.
[24] J. Zhang, Y. Kawai, T. Kumamoto, and K. Tanaka, "A novel visualization method for distinction of web news sentiment," in *Proc. 10th Int. Conf. Web Inf. Syst. Eng.*, 2009, pp. 181–194.
[25] W. S. Cleveland and C. L. Loader, *Smoothing by Local Regression: Principles and Methods*. New York, NY, USA: Springer, pp. 10–49, 1996.

**Mikalai Tsytsarau** received the master's degree from Belarusian State University and the PhD degree from the Department of Information Engineering and Computer Science, DISI, University of Trento. He is a researcher with the Database and Information Management Group dbTrento. He worked with Yahoo! Research and HP Labs as a visiting researcher and with Qatar Computing Research Institute as a research associate.

**Themis Palpanas** is a professor at the Paris Descartes University. He has previously held positions at the University of Trento, IBM T.J. Watson Research Center, the University of California at Riverside, and has been a visiting researcher at the National University of Singapore, the IBM Almaden Research Center, and Microsoft Research. He is a founding member of the Event Processing Technical Society, editor in chief of the *Big Data Research Journal*, associate editor of the *IEEE Transactions on Knowledge and Data Engineering*, and editorial board member of IS and IDA journals. He was a general chair for VLDB 2013.