Scalable Discovery of Contradictions on the Web

Mikalai Tsytsarau
University of Trento
Trento, Italy
tsytsarau@disi.unitn.eu

Themis Palpanas
University of Trento
Trento, Italy
themis@disi.unitn.eu

Kerstin Denecke
L3S Research Center
Hannover, Germany
denecke@L3S.de

ABSTRACT

Our study addresses the problem of large-scale contradiction detection and management, from data extracted from the Web. We describe the first systematic solution to the problem, based on a novel statistical measure for contradictions, which exploits first- and second-order moments of sentiments. Our approach enables the interactive analysis and online identification of contradictions under multiple levels of time granularity. The proposed algorithm can be used to analyze and track opinion evolution over time and to identify interesting trends and patterns. It uses an incrementally updatable data structure to achieve computational efficiency and scalability. Experiments with real datasets show promising time performance and accuracy.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Performance

1. INTRODUCTION

Contradiction analysis is a newly emerged research area, where we are interested in automatically discovering topics for which different opinions have been expressed across space (i.e., users) and time. Relevant previous work has appeared in the areas of Natural Language Processing [3, 4] and Opinion Mining [1, 5]. The focus of the latter studies is how to effectively track the evolution of opinions inside different communities (like weblogs, or social networks), where diverse opinions are very common [2, 6]. Furthermore, the existing opinion aggregation methods are not designed specifically for the contradiction detection or do not preserve enough information to drive the subsequent contradiction analysis step. The work by Chen et al. [1] is the one most closely related to our problem. In their solution, they produce graphs that have to be visually inspected in order to identify the contradictions.

In contrast, we are proposing the first systematic approach on aggregating opinions with respect to some topic, along with the necessary mechanisms for achieving a reliable and computationally efficient solution for identifying contradictions across different texts and time granularities. Our approach also allows the identification of two different types of contradictions, namely, overlapping contradicting opinions (*simultaneous contradiction*), and opinions that change average polarity at some point in time (*change of sentiment*). We designed our contradiction data structure to be space-efficient, incrementally updatable, and scalable both on the number of topics and posts. As we discuss in more detail later, the performance evaluation shows that our solution helps answer contradiction queries in a large scale more than two orders of magnitude faster than a relational database.

2. PROPOSED APPROACH

In this work, we use the continuous range of [-1;1] to represent topic-wise sentiment values S. This gives us flexibility in using external sentiment extraction methods and allows to perform simple and straightforward aggregation. For the rest of this paper, we assume that we analyze sentiments on some predefined topic T, over a collection of postings or texts $\mathcal{P}$.

In order to be able to identify contradicting opinions we define a novel measure of contradiction. The intuition behind this measure is that when the aggregated value for sentiments (on a specific topic and time interval) is closer to zero, while the sentiment diversity is high, then the contradiction should be high. Accordingly, we define the *Aggregated Sentiment* μ_S as the mean value over all individual sentiments, and *Sentiment Diversity* σ_S^2 as their variance:

$$\mu_S = \frac{1}{n}\sum_{i=1}^{n} S_i, \qquad \sigma_S^2 = \frac{1}{n}\sum_{i=1}^{n}(S_i - \mu_S)^2, \tag{1}$$

where n is the cardinality of $\mathcal{P}$. Evidently, we need to combine μ_S and σ_S^2 in a single formula for computing contradictions. We propose the following formula for contradictions:

$$C = \frac{\vartheta \cdot \sigma_S^2}{\vartheta + (\mu_S)^2} W \tag{2}$$

In the denominator, we add a small value, $\vartheta \neq 0$, which allows to limit the level of contradiction C when $(\mu_S)^2$ is close to zero. The nominator is multiplied by ϑ to ensure that contradiction values fall within the interval $[0; 1]$. W is a weight function aiming to compensate the contradiction value for the varying number of posts that may be involved in the calculation:

$$W = (1 + exp(\frac{\overline{n} - n}{\beta}))^{-1} \tag{3}$$

where the constant $\overline{n}$ reflects the average number of texts in the collection, and β is a scaling factor. This weight func-

Copyright is held by the author/owner(s).
WWW 2010, April 26–30, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

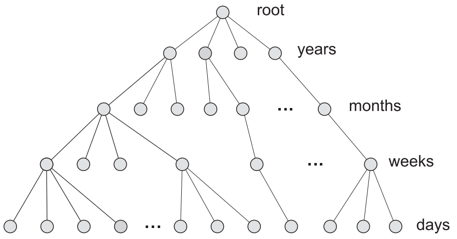

Figure 1: Logical representation of the Contradiction Tree.

tion provides a multiplicative factor in the range [0; 1]. Our experiments show that the contradiction measure is rather effective even with the default values of these parameters (in this work, we use $\vartheta = 5 \cdot 10^{-4}$, $\overline{n} = 30$ and $\beta = 10$).

2.1 Storing Contradiction Values

Although there exist several possible ways of organizing the data, we propose to store contradiction values for different topics under the same time-tree structure, which we call the Contradiction Tree (CTree). It is organized around the aggregated moments of sentiments, and a hierarchical segmentation of time, as shown in Figure 1.

Using this data structure, not only can we answer queries on *adhoc* time intervals, by dynamically computing the contradiction values, but we can also incrementally update the information stored in the CTree. This is true, because our contradiction measure is based on the mean and variance of the sentiments, which can be computed using the first- and second-order moments of sentiments. The latter are updatable and can be aggregated over various time intervals.

2.2 Answering Contradiction Queries

When detecting contradictions, we can set some fixed threshold ρ and report only the time intervals having contradiction values above ρ. We refer to this solution as *fixed threshold*. Alternatively, we can use an *adaptive threshold* technique, which can better fit the nature of the data within each time window (that may vary over time and across topics). In this case, we compute a different threshold for each time window, based on the contradiction value of its parent.

3. EXPERIMENTAL EVALUATION

We evaluated the effectiveness of the contradiction measure using two real datasets, i.e., medical blogs (webmd.com) and political commentaries (slashdot.org). For brevity, we only report some results from the Slashdot dataset, for the topic ”internet government control”. Our analysis identified three major contradictions (marked 1-3 in the bottom graph of the Figure 2), all discussing the pros and cons of a law that would give the government more power in controlling the internet traffic (table 1 shows extracts from opposing posts that contributed to contradiction 2). Evidently, these are all very relevant discussions that express different points of view on the same topic, but are not easy to identify with a quick visual inspection of the raw sentiments. Thus, having an automated way of identifying them can be very useful.

We also compare the scalability of contradiction detection using a CTree to a relational database implementation. We generated a synthetic dataset of 80 million random sentiments for 10,000 topics over a time interval of 4 years, and used 25 queries that extract contradictions at random granularities and time intervals (on all topics).

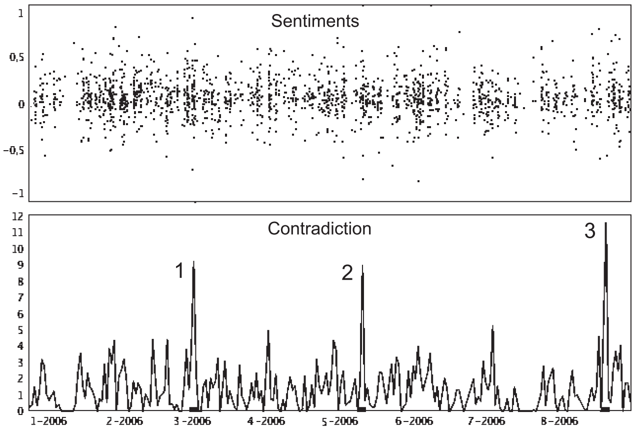

Figure 2: Raw sentiment and contradiction values.

topic ”internet government control”, Slashdot
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.
PRO: I suppose we better wrap a firewall around our country and not let those damn foreigners access to our internet.
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.
CON: While it sounds like a decent idea, I’m really all for the whole uncensored and unregulated internet.

Table 1: Examples of contradicting posts.

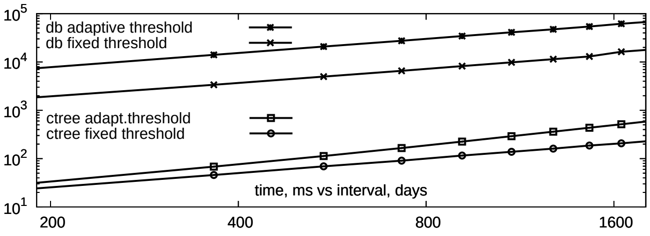

Figure 3: Scalability of CTree and DB approaches.

Figure 3 shows the time needed to execute these queries for both the fixed and the adaptive thresholds. The adaptive threshold queries require in all cases more time since the threshold computation depends on the contradiction value of the parent time window. We observe that all queries scale linearly with the size of time interval. The results also show that the CTree approach performs 2 (in some cases almost 3) orders of magnitude faster than the database solution.

4. REFERENCES

[1] 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, ’06.

[2] 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.

[3] M. C. de Marneffe, A. N. Rafferty, and C. D. Manning. Finding contradictions in text. In *Proceedings of ACL-08: HLT*, pages 1039–1047, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[4] S. Harabagiu, A. Hickl, and F. Lacatusu. Negation, contrast and contradiction in text processing. In *AAAI’06: Proceedings of the 21st national conference on Artificial intelligence*, pages 755–762. AAAI Press, 2006.

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

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