Diversity-Aware Clustering of SIOC Posts

Andreas Thalhammer, Ioannis Stavrakantonakis, and Ioan Toma

University of Innsbruck, Technikerstr. 21a, A-6020 Innsbruck
{andreas.thalhammer, ioannis.stavrakantonakis, ioan.toma}@sti2.at

Abstract. Sentiment analysis as well as topic extraction and named entity recognition are emerging methods used in the field of Web Mining. Next to SQL-like querying and according visualization, new ways of organization have become possible. In this demo paper we apply efficient clustering algorithms that stem from the image retrieval field to `sioc:Post` entities, blending similarity scores of sentiment and covered topics. We demonstrate the system with a visualization component that combines different diversity aspects within microposts by Twitter users and a static news article collection.

1 Introduction

Named entity recognition, automatic tagging, and sentiment detection in microposts, news articles, blog posts, forum posts etc. provide us new ways of interacting with content. Not only is it possible to retrieve answers from queries like *"select all positive articles that mention Barack Obama"* but these features offer a new way of content organization: combining sentiment and topic similarity in a single clustering approach. This enables the user to browse datasets in a novel way, for example getting overviews on positive and negative opinions on the topic *"champions league final"* or retrieving different topic clusters in negative Tweets from a specific user.

In this work, we demonstrate the application of two efficient clustering algorithms that stem from the image retrieval domain to sentiment analysis in combination with topic extraction and named entity recognition. We apply our approach on two use cases: microposts and news articles. Moreover, the readers are invited to try the system with live Twitter data to find new insights about the polarity and topic distribution of politicians' Tweets as well as their own.

2 Related Work

The contribution of our work is twofold, from a cluster dimension perspective (i.e., sentiment and topics are covered) as well as from a domain perspective (i.e., news articles and Tweets are covered). In this short paper we are not able to provide an extensive overview of the state of the art but we would like to contextualize our approach along with two related approaches.

[3] presents a study on automatically clustering and classifying Tweets. The outcomes of the paper stress out that employing a supervised methodology based

on hash-tags could produce better results than the traditional unsupervised methods. Furthermore, the authors present a methodology for finding the most representative Tweet in a cluster. Automatic detection of topics discussed in Tweets is pointed out as one of the interesting problems in Tweet analysis.

[2] proposes an emotion-oriented clustering approach in accordance to sentiment similarities between blog search result titles and snippets. The authors propose an approach for grouping blog search results in sentiment clusters, which is related to the grouping that we perform in the retrieved articles when we choose to cluster them based on the sentiment rather than the topic. The authors' goals are similar to ours as the approach focuses on very short text portions, which is also covered by our method as we cluster Tweets which are no longer than 140 characters. The sentiment detection relies on the SentiWordNet[1] which is built on top of WordNet and it provides sentiment scores on the glosses of WordNet.

In comparison to [3] and [2] which focus on clustering either by topics or sentiments, our approach combines those elements in a flexible way. For this, we introduce a straight-forward combination of topic and sentiment similarity measures that can be flexibly adapted to be more specific towards either topic or sentiment. Similarly to [2] we try to cover clusters of microposts as well as longer articles.

3 Data Extraction, Modeling, and Storage

We utilize the Twitter API to access the microposts and a static news corpus of the RENDER project[2]. The extracted Twitter data is processed using the Enrycher service[3] and stored in a Sesame[4] or OWLIM[5] triple store. The news data is already processed with Enrycher and already available in the correct format in an OWLIM triple store. As a data model we are utilizing the sioc [1] vocabulary in combination with the Knowledge Diversity Ontology[6] (KDO) [4]. KDO was developed in the context of the RENDER project and features assigning sentiments to sioc posts. Moreover we make use of the newly introduced type `kdo:NewsArticle` and the class `sioc-types:MicroblogPost`, both being subclasses of `sioc:Post`. In accordance to the respective document, the Enrycher service [6] assigns to instances of these subclasses a range of `sioc:topic`s as well as a sentiment (i.e., `kdo:hasSentiment`). The data model as well as instances are stored in and retrieved from a triple store implementing the SAIL[7] interface (e.g. OWLIM).

[1] SentiWordNet – `http://sentiwordnet.isti.cnr.it/`
[2] RENDER News Corpus – `http://rendernews.ontotext.com/`, RENDER project – `http://render-project.eu`
[3] Enrycher – `http://enrycher.ijs.si`, `http://ailab.ijs.si/tools/enrycher/`
[4] Sesame - `http://www.openrdf.org/`
[5] OWLIM – `http://owlim.ontotext.com/`
[6] KDO – `http://kdo.render-project.eu/`
[7] SAIL API – `http://www.openrdf.org/doc/sesame2/system/ch05.html`

4 Diversity-Aware Clustering

Van Leuken et al. introduce "visual diversification of image search results" in [5]. The involved clustering algorithms are reported to be effective and efficient. The introduced similarity measures are based on visual similarity of images. For our document-based approach, we employ a combination of two similarity measures, namely topic and sentiment similarity. The final score is calculated with a flexible weighting component γ (with $0 \leq \gamma \leq 1$). We calculate the similarity of two `sioc:Posts` p_1 and p_2 as follows:

$$sim(p_1, p_2) = \gamma \cdot jacc(p_1, p_2) + (1 - \gamma) \cdot sent(p_1, p_2) \tag{1}$$

In formula 1 the functions *jacc* and *sent* need yet to be defined. *jacc* is basically a simple Jaccard similarity index over topics:

$$jacc(p_1, p_2) = \frac{|topics(p_1) \cap topics(p_2)|}{|topics(p_1) \cup topics(p_2)|} \tag{2}$$

We assume the extracted sentiment scores to be in the interval of $[0, 1]$ with 1 being most positive and 0 being most negative. The similarity score *sent* takes this into account, having the highest similarity of 1 if the two scores are equal. This similarity score is calculated as follows:

$$sent(p_1, p_2) = 1 - |score(p_1) - score(p_2)| \tag{3}$$

For the case that the scores are not in the mentioned interval, they are normalized as follows:

$$score(p) = \frac{score(p) - min(score(p))}{max(score(p)) - min(score(p))} \tag{4}$$

We utilize the Folding and Maximum algorithm from [5]. These algorithms were originally designed to cluster in accordance to visual similarity of images. Rather than using image histograms, we apply these algorithms to textual features of posts, using the similarity measure from above (see Formula 1).

The Folding algorithm assumes a ranked list as input. There are two disjoint lists maintained, the representatives and the rest. At the start, the ranked input is the rest. The algorithm selects the first element of the rest (i.e., the ranked input list) as a representative. In the following, each element of the rest is compared to the representatives and added to the representatives list in case its similarity to all existing representatives is less than a certain reference point (i.e., a variable ϵ). When all representatives are established, each element in the rest is assigned to the cluster of which the representative is most similar to it.

The Maximum algorithm is similar to Folding but has some distinct features. The Maximum algorithm belongs to the class of randomized algorithms. Again there are two disjoint lists, the representatives and the rest which is assigned to the input at the beginning. The first element of the representatives is selected randomly from the rest. Then, the algorithm adds the element which

```
Data: List L containing sioc posts
Result: double value of ϵ
sumAll := 0;
for each sioc:Post s1 in L do
    sum := 0;
    for each sioc:Post s2 in L do
        if s1 != s2 then
            Sum := Sum + sim(s1, s2);
    Avg := Sum / (size(L) -1);
    SumAll := SumAll + Avg;
return SumAll / size(L);
```

Algorithm 1: ϵ estimation

has minimum maximum similarity (or maximum minimum distance) to the representatives. If this minimum maximum similarity is at some point less than ϵ, all representatives are found and the remaining elements in the rest list are assigned to the clusters with closest representatives.

Both algorithms produce clusters, each with a selected representative. However, as a last point, it remains open how to select an appropriate value for ϵ. In this step we determine the average similarity of a `sioc:Post` to another (see Algorithm 1).

5 Implementation

We implemented the diversity-aware ranking service with Oracle GlassFish 3.x. The source code is available as a github project[8] and a deployment can be found at `http://ranking.render-project.eu/`. There, users can specify a variety of parameters and retrieve the JSON output for the clustering. For a better user experience, we introduce a jQuery-based visualization component that is demonstrated at `http://ranking.render-project.eu/tweetVis.html` (Twitter) and `http://ranking.render-project.eu/vis.html` (news). Figure 1 shows the news visualization component. The slider at the top changes the γ value of the similarity measure (see Formula 1) either towards sentiment similarity or topic similarity.

6 Conclusion

We have implemented a diversity-aware ranking service that enables clustering and retrieval of sioc posts along the two dimensions: sentiment and topic. We exemplify our approach on live Twitter data and a static news dataset. This work is also meant to initiate new directions to look at content organization, navigation, and presentation.

[8] Source code – `https://github.com/athalhammer/RENDER-ranking-service`

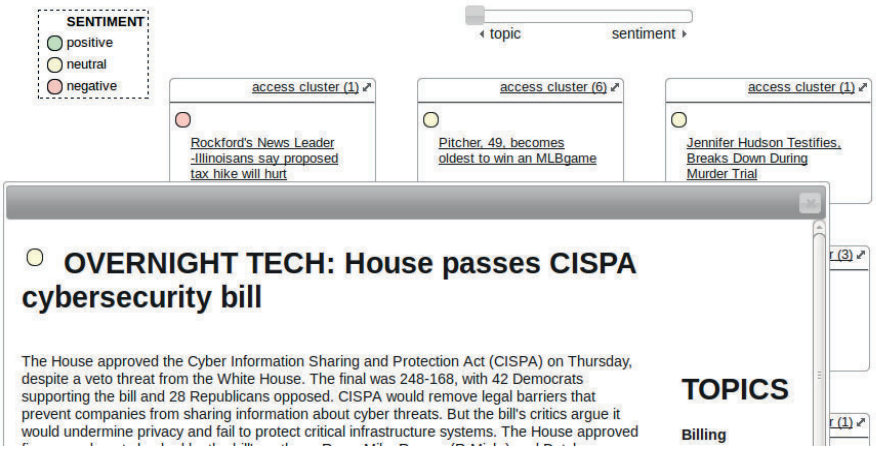

Fig. 1. The news visualization component.

Acknowledgements We are grateful for the feedback from Daniele Pighin (Google Zurich). This research was partly funded by the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257790 (RENDER project).

References

1. John G. Breslin, Andreas Harth, Uldis Bojars, and Stefan Decker. Towards semantically-interlinked online communities. In *The Semantic Web: Research and Applications*, volume 3532 of *Lecture Notes in Computer Science*, pages 500–514. Springer Berlin Heidelberg, 2005.
2. Shi Feng, Daling Wang, Ge Yu, Chao Yang, and Nan Yang. Sentiment clustering: A novel method to explore in the blogosphere. In *Proceedings of the Joint International Conferences on Advances in Data and Web Management*, APWeb/WAIM '09, pages 332–344, Berlin, Heidelberg, 2009. Springer-Verlag.
3. K. D. Rosa, R. Shah, B. Lin, A. Gershman, and R. Frederking. Topical clustering of tweets. *Proceedings of the ACM SIGIR: SWSM*, 2011.
4. Andreas Thalhammer, Ioan Toma, Rakebul Hasan, Elena Simperl, and Denny Vrandečić. How to represent knowledge diversity. Poster at 10th intl. Semantic Web Conf. (ISWC11), 10 2011.
5. Reinier H. van Leuken, Lluis Garcia, Ximena Olivares, and Roelof van Zwol. Visual diversification of image search results. In *Proc. of the 18th intl. conf. on World Wide Web*, WWW '09, pages 341–350, New York, NY, USA, 2009. ACM.
6. Tadej Štajner, Delia Rusu, Lorand Dali, Balž Fortuna, Dunja Mladenić, and Marko Grobelnik. Enrycher: service oriented text enrichment. In *Proc. of the 11th intl. multiconference Information Society*, IS-2009, 2009.