

Subgroup Detector: A System for Detecting Subgroups in Online Discussions

Amjad Abu-Jbara
EECS Department
University of Michigan
Ann Arbor, MI, USA
amjbara@umich.edu

Dragomir Radev
EECS Department
University of Michigan
Ann Arbor, MI, USA
radev@umich.edu

Abstract

We present Subgroup Detector, a system for analyzing threaded discussions and identifying the attitude of discussants towards one another and towards the discussion topic. The system uses attitude predictions to detect the split of discussants into subgroups of opposing views. The system uses an unsupervised approach based on rule-based opinion target detecting and unsupervised clustering techniques. The system is open source and is freely available for download. An online demo of the system is available at: <http://clair.eecs.umich.edu/SubgroupDetector/>

1 Introduction

Online forums discussing ideological and political topics are common¹. When people discuss a controversial topic, it is normal to see situations of both agreement and disagreement among the discussants. It is even not uncommon that the big group of discussants split into two or more smaller subgroups. The members of each subgroup have the same opinion toward the discussion topic. The member of a subgroup is more likely to show positive attitude to the members of the same subgroup, and negative attitude to the members of opposing subgroups. For example, consider the following snippet taken from a debate about school uniform

¹www.politicalforum.com, www.createdebate.com, www.forandagainst.com, etc

(1) Discussant 1: I believe that school uniform is a good idea because it improves student attendance.

(2) Discussant 2: I disagree with you. School uniform is a bad idea because people cannot show their personality.

In (1), the writer is expressing positive attitude regarding school uniform. The writer of (2) is expressing negative attitude (disagreement) towards the writer of (1) and negative attitude with respect to the idea of school uniform. It is clear from this short dialog that the writer of (1) and the writer of (2) are members of two opposing subgroups. Discussant 1 supports school uniform, while Discussant 2 is against it.

In this demo, we present an unsupervised system for determining the subgroup membership of each participant in a discussion. We use linguistic techniques to identify attitude expressions, their polarities, and their targets. We use sentiment analysis techniques to identify opinion expressions. We use named entity recognition, noun phrase chunking and coreference resolution to identify opinion targets. Opinion targets could be other discussants or subtopics of the discussion topic. Opinion-target pairs are identified using a number of hand-crafted rules. The functionality of this system is based on our previous work on attitude mining and subgroup detection in online discussions.

This work is related to previous work in the areas of sentiment analysis and online discussion mining. Many previous systems studied the problem of iden-

tifying the polarity of individual words (Hatzivasiloglou and McKeown, 1997; Turney and Littman, 2003). Opinionfinder (Wilson et al., 2005) is a system for mining opinions from text. SENTIWORDNET (Esuli and Sebastiani, 2006) is a lexical resource in which each WordNet synset is associated to three numerical scores Obj(s), Pos(s) and Neg(s), describing how objective, positive, and negative the terms contained in the synset are. Dr Sentiment (Das and Bandyopadhyay, 2011) is an online interactive gaming technology used to crowd source human knowledge to build an extension of SentiWordNet.

Another research line focused on analyzing online discussions. For example, Lin et al. (2009) proposed a sparse coding-based model that simultaneously models the semantics and the structure of threaded discussions. Shen et al. (2006) proposed a method for exploiting the temporal and lexical similarity information in discussion streams to identify the reply structure of the dialog. Many systems addressed the problem of extracting social networks from discussions (Elson et al., 2010; McCallum et al., 2007). Other related sentiment analysis systems include MemeTube (Li et al., 2011), a sentiment-based system for analyzing and displaying microblog messages; and C-Feel-It (Joshi et al., 2011), a sentiment analyzer for micro-blogs.

In the rest of this paper, we describe the system architecture, implementation, usage, and its evaluation.

2 System Overview

Figure 1 shows a block diagram of the system components and the processing pipeline. The first component is the *thread parsing* component which takes as input a discussion thread and parses it to identify posts, participants, and the reply structure of the thread. The second component in the pipeline processes the text of posts to identify polarized words and tag them with their polarity. The list of polarity words that we use in this component has been taken from the OpinionFinder system (Wilson et al., 2005).

The polarity of a word is usually affected by the context in which it appears. For example, the word *fine* is positive when used as an adjective and negative when used as a noun. For another example, a positive word that appears in a negated context becomes negative. To address this, we take the part-of-speech (POS) tag of the word into consideration when we assign word polarities. We require that the POS tag of a word matches the POS tag provided in the list of polarized words that we use. The negation issue is handled in the opinion-target pairing step as we will explain later.

The next step in the pipeline is to identify the candidate targets of opinion in the discussion. The target of attitude could be another discussant, an entity mentioned in the discussion, or an aspect of the discussion topic. When the target of opinion is another discussant, either the discussant name is mentioned explicitly or a second person pronoun (e.g you, your, yourself) is used to indicate that the opinion is targeting the recipient of the post.

The target of opinion could also be a subtopic or an entity mentioned in the discussion. We use two methods to identify such targets. The first method depends on identifying noun groups (NG). We consider as an entity any noun group that is mentioned by at least two different discussants. We only consider as entities noun groups that contain two words or more. We impose this requirement because individual nouns are very common and considering all of them as candidate targets will introduce significant noise. In addition to this shallow parsing method, we also use named entity recognition (NER) to identify more targets. The named entity tool that we use recognizes three types of entities: person, location, and organization. We impose no restrictions on the entities identified using this method.

A challenge that always arises when performing text mining tasks at this level of granularity is that entities are usually expressed by anaphorical pronouns. Jakob and Gurevych (2010) showed experimentally that resolving the anaphoric links

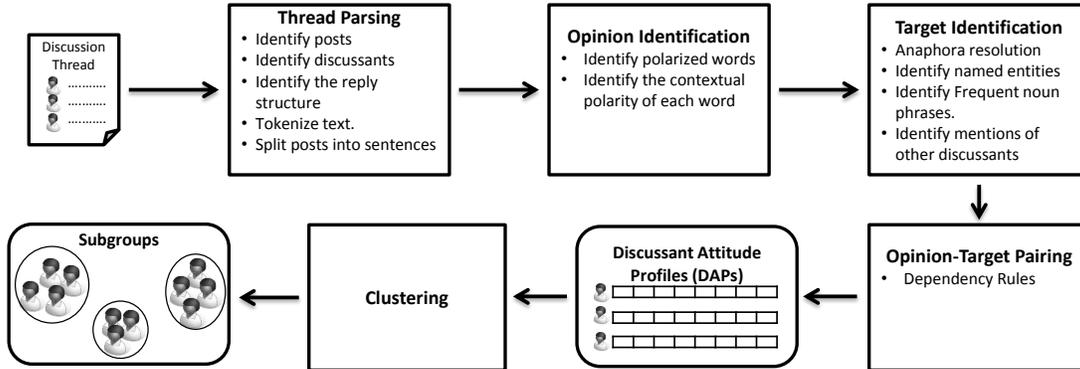


Figure 1: A block diagram illustrating the processing pipeline of the subgroup detection system

in text significantly improves opinion target extraction. Therefore, we use co-reference resolution techniques to resolve all the anaphoric links in the discussion thread.

At this point, we have all the opinion words and the potential targets identified separately. The next step is to determine which opinion word is targeting which target. We propose a rule based approach for opinion-target pairing. Our rules are based on the dependency relations that connect the words in a sentence. An opinion word and a target form a pair if the dependency path between them satisfies at least one of our dependency rules. Table 1 illustrates some of these rules. The rules basically examine the types of dependency relations on the shortest path that connect the opinion word and the target in the dependency parse tree. It has been shown in previous work on relation extraction that the shortest dependency path between any two entities captures the information required to assert a relationship between them (Bunescu and Mooney, 2005). If a sentence S in a post written by participant P_i contains an opinion word OP_j and a target TR_k , and if the opinion-target pair satisfies one of our dependency rules, we say that P_i expresses an attitude towards TR_k . The polarity of the attitude is determined by the polarity of OP_j . We represent this as $P_i \xrightarrow{+} TR_k$ if OP_j is positive and $P_i \xrightarrow{-} TR_k$ if OP_j is negative. Negation is handled in this step by reversing the polarity if the polarized expression is part of a

neg dependency relation.

It is likely that the same participant P_i expresses sentiment towards the same target TR_k multiple times in different sentences in different posts. We keep track of the counts of all the instances of positive/negative attitude P_i expresses toward TR_k . We represent this as $P_i \xrightarrow[m-]{m+} TR_k$ where m (n) is the number of times P_i expressed positive (negative) attitude toward TR_k .

Now, we have information about each discussant attitude. We propose a representation of discussants attitudes towards the identified targets in the discussion thread. As stated above, a target could be another discussant or an entity mentioned in the discussion. Our representation is a vector containing numerical values. The values correspond to the counts of positive/negative attitudes expressed by the discussant toward each of the targets. We call this vector the *discussant attitude profile (DAP)*. We construct a DAP for every discussant. Given a discussion thread with d discussants and e entity targets, each attitude profile vector has $n = (d + e) * 3$ dimensions. In other words, each target (discussant or entity) has three corresponding values in the DAP: 1) the number of times the discussant expressed positive attitude toward the target, 2) the number of times the discussant expressed a negative attitude towards the target, and 3) the number of times the the discussant interacted with or mentioned the target. It has to be noted that these values are not symmet-

ID	Rule	In Words
R1	$OP \rightarrow nsubj \rightarrow TR$	The target TR is the nominal subject of the opinion word OP
R2	$OP \rightarrow dobj \rightarrow TR$	The target T is a direct object of the opinion OP
R3	$OP \rightarrow prep_* \rightarrow TR$	The target TR is the object of a preposition that modifies the opinion word OP
R4	$TR \rightarrow amod \rightarrow OP$	The opinion is an adjectival modifier of the target
R5	$OP \rightarrow nsubjpass \rightarrow TR$	The target TR is the nominal subject of the passive opinion word OP
R6	$OP \rightarrow prep_* \rightarrow poss \rightarrow TR$	The opinion word OP connected through a $prep_*$ relation as in $R2$ to something possessed by the target TR
R7	$OP \rightarrow dobj \rightarrow poss \rightarrow TR$	The target TR possesses something that is the direct object of the opinion word OP
R8	$OP \rightarrow csubj \rightarrow nsubj \rightarrow TR$	The opinion word OP is a causal subject of a phrase that has the target TR as its nominal subject.

Table 1: Examples of the dependency rules used for opinion-target pairing.

ric since the discussions explicitly denote the source and the target of each post.

At this point, we have an attitude profile (or vector) constructed for each discussant. Our goal is to use these attitude profiles to determine the subgroup membership of each discussant. We can achieve this goal by noticing that the attitude profiles of discussants who share the same opinion are more likely to be similar to each other than to the attitude profiles of discussants with opposing opinions. This suggests that clustering the attitude vector space will achieve the goal and split the discussants into subgroups based on their opinion.

3 Implementation

The system is fully implemented in Java. Part-of-speech tagging, noun group identification, named entity recognition, co-reference resolution, and dependency parsing are all computed using the Stanford Core NLP API.² The clustering component uses the JavaML library³ which provides implementations to several clustering algorithms such as k-means, EM, FarthestFirst, and OPTICS.

The system requires no installation. It, however, requires that the Java Runtime Environment (JRE) be installed. All the dependencies of the system come bundled with the system in the same package. The system works on all the standard platforms.

The system has a command-line interface that

provides full access to the system functionality. It can be used to run the whole pipeline to detect subgroups or any portion of the pipeline. For example, it can be used to tag an input text with polarity or to identify candidate targets of opinion in a given input. The system behavior can be controlled by passing arguments through the command line interface. For example, the user can specify which clustering algorithm should be used.

To facilitate using the system for research purposes, the system comes with a clustering evaluation component that uses the ClusterEvaluator package.⁴ If the input to the system contains subgroup labels, it can be run in the evaluation mode in which case the system will output the scores of several different clustering evaluation metrics such as purity, entropy, f-measure, Jaccard, and RandIndex. The system also has a Java API that can be used by researchers to develop other systems using our code.

The system can process any discussion thread that is input to it in a specific format. The format of the input and output is described in the accompanying documentation. It is the user responsibility to write a parser that converts an online discussion thread to the expected format. However, the system package comes with two such parsers for two different discussion sites: www.politicalforum.com and www.createdebate.com.

The distribution also comes with three datasets

²<http://nlp.stanford.edu/software/corenlp.shtml>

³<http://java-ml.sourceforge.net/>

⁴<http://eniac.cs.qc.cuny.edu/andrew/v-measure/javadoc/index.html>

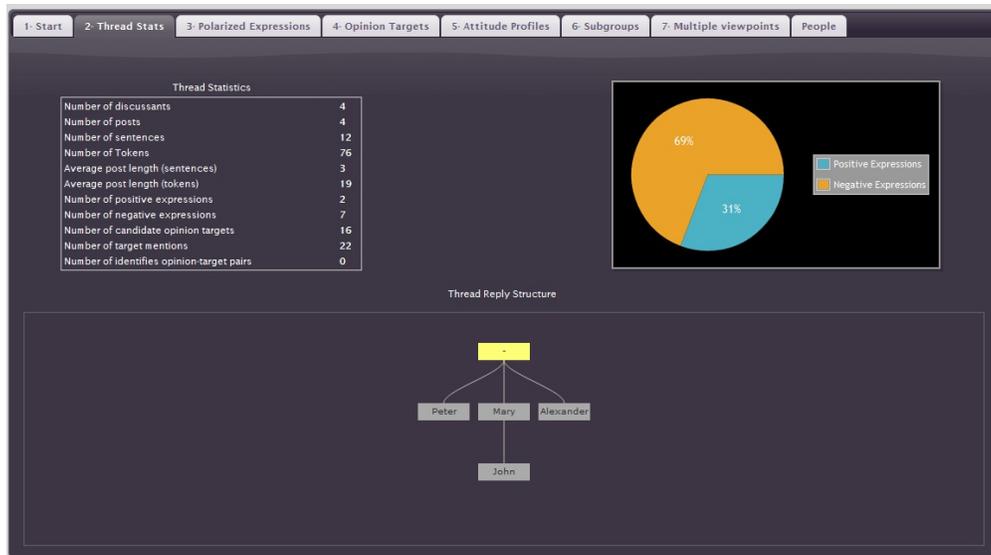


Figure 2: A screenshot of the online demo

(from three different sources) comprising a total of 300 discussion threads. The datasets are annotated with the subgroup labels of discussants.

Finally, we created a web interface to demonstrate the system functionality. The web interface is intended for demonstration purposes only. No web-service is provided. Figure 2 shows a screenshots of the web interface. The online demo can be accessed at <http://clair.eecs.umich.edu/SubgroupDetector/>

4 Evaluation

In this section, we give a brief summary of the system evaluation. We evaluated the system on discussions comprising more than 10,000 posts in more than 300 different topics. Our experiments show that the system detects subgroups with promising accuracy. The average clustering purity of the detected subgroups in the dataset is 0.65. The system significantly outperforms baseline systems based on text clustering and discussant interaction frequency. Our experiments also show that all the components in the system (such as co-reference resolution, noun phrase chunking, etc) contribute positively to the accuracy.

5 Conclusion

We presented a demonstration of a discussion mining system that uses linguistic analysis techniques to predict the attitude the participants in online discussions forums towards one another and towards the different aspects of the discussion topic. The system is capable of analyzing the text exchanged in discussions and identifying positive and negative attitudes towards different targets. Attitude predictions are used to assign a subgroup membership to each participant using clustering techniques. The system predicts attitudes and identifies subgroups with promising accuracy.

References

- Razvan Bunescu and Raymond Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.
- Amitava Das and Sivaji Bandyopadhyay. 2011. Dr sentiment knows everything! In *Proceedings of the ACL-HLT 2011 System Demonstrations*, pages 50–55, Portland, Oregon, June. Association for Computational Linguistics.

- David Elson, Nicholas Dames, and Kathleen McKeown. 2010. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147, Uppsala, Sweden, July.
- Andrea Esuli and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC06)*, pages 417–422.
- Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *EACL'97*, pages 174–181.
- Niklas Jakob and Iryna Gurevych. 2010. Using anaphora resolution to improve opinion target identification in movie reviews. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 263–268, Uppsala, Sweden, July. Association for Computational Linguistics.
- Aditya Joshi, Balamurali AR, Pushpak Bhattacharyya, and Rajat Mohanty. 2011. C-feel-it: A sentiment analyzer for micro-blogs. In *Proceedings of the ACL-HLT 2011 System Demonstrations*, pages 127–132, Portland, Oregon, June. Association for Computational Linguistics.
- Cheng-Te Li, Chien-Yuan Wang, Chien-Lin Tseng, and Shou-De Lin. 2011. Memetube: A sentiment-based audiovisual system for analyzing and displaying microblog messages. In *Proceedings of the ACL-HLT 2011 System Demonstrations*, pages 32–37, Portland, Oregon, June. Association for Computational Linguistics.
- Chen Lin, Jiang-Ming Yang, Rui Cai, Xin-Jing Wang, and Wei Wang. 2009. Simultaneously modeling semantics and structure of threaded discussions: a sparse coding approach and its applications. In *SIGIR '09*, pages 131–138.
- Andrew McCallum, Xuerui Wang, and Andrés Corrada-Emmanuel. 2007. Topic and role discovery in social networks with experiments on enron and academic email. *J. Artif. Int. Res.*, 30:249–272, October.
- Dou Shen, Qiang Yang, Jian-Tao Sun, and Zheng Chen. 2006. Thread detection in dynamic text message streams. In *SIGIR '06*, pages 35–42.
- Peter Turney and Michael Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21:315–346.
- Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Opinionfinder: a system for subjectivity analysis. In *HLT/EMNLP - Demo*.