Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM 2013)

August 11, 2013, Chicago, IL, USA

held in conjunction with

SIGKDD 2013

Workshop Organizers

Erik Cambria, National University of Singapore (Singapore)

Bing Liu, University of Illinois at Chicago (USA)

Yongzheng Zhang, eBay Inc. (USA)

Yunqing Xia, Tsinghua University (China)

The Association for Computing Machinery, Inc.
1515 Broadway
New York, New York 10036

ACM ISBN: 978-1-4503-2332-1

Additional copies may be ordered prepaid from:

ACM Order Department
P.O. BOX 11405
Church Street Station
New York, NY 10286-1405

Phone: 1-800-342-6626
(U.S.A. and Canada)
+1-212-626-0500
(All other countries)
Fax: +1-212-944-1318
E-mail: acmhelp@acm.org

Printed in the U.S.A.

Remarks

The exponential growth of the Social Web is virally infecting more and more critical business processes such as customer support and satisfaction, brand and reputation management, product design and marketing. Because of this global trend, web users already evolved from the era of social relationships, in which they began to get connected and started to share contents, to the era of social functionality, in which they started using social networks as the main platform for communication and dissemination of information. Today, web users are going through the era of social colonization, in which every experience on the Web can be social (e.g., Facebook Like button), and are getting ready for the era of social context, in which web contents will be highly targeted and personalized. The final stage of such Social Web evolution is the so called era of social commerce, in which communities will define future products and services. In such context, the research field of sentiment analysis, which has already been rapidly growing in the last decade, is destined to become more and more important for Web and business dynamics. To this end, the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM 2013: http://sentic.net/wisdom) aims to explore how the wisdom of the crowds is affecting (and will affect) the evolution of the Web and of businesses gravitating around it. In particular, the workshop explores two different stages of sentiment analysis: the former focusing on the identification of opinionated text over the Web, the latter focusing on the classification of such text either in terms of polarity detection or emotion recognition.

Topics of Interest

The workshop will provide an international forum for both researchers and entrepreneurs working in the field of opinion mining to share information on their latest investigations in social information retrieval and their applications in academic research areas and industrial sectors. The broader context of the workshop comprehends AI, Semantic Web, information retrieval, web mining, and natural language processing (NLP). In addition to paper presentations, an invited talk by Professor Bing Liu will stress the problem of detecting fake opinions in social media. Topics of interest include but are not limited to:

- Sentiment identification & classification
- Knowledge-based opinion mining
- Sentiment summarization & visualization
- Entity discovery & extraction
- Opinion aggregation
- Opinion search & retrieval
- Time evolving sentiment analysis
- Opinion spam detection
- Comparative opinion analysis
- Topic detection & trend discovery
- Psychological models for sentiment analysis
- Multilingual opinion mining
- Social ranking
- Social network analysis
- Influence, trust & privacy analysis
- Business intelligence applications

Program Schedule

09:00-10:10:

- **Opening Remarks**
- **Keynote Talk: Statistical Methods for Integration and Analysis of Opinionated Text Data**
  Chengxiang Zhai, University of Illinois at Urbana-Champaign

10:10-10:30: Coffee Break

10:30-12:00: Session I

- **Identifying Purpose Behind Electoral Tweets**
  Saif M. Mohammad, Svetlana Kiritchenko, and Joel Martin
- **Combining Strengths, Emotions and Polarities for Boosting Twitter Sentiment Analysis**
  Felipe Bravo-Marquez, Marcelo Mendoza, and Barbara Poblete
- **Modelling Political Disaffection from Twitter Data**
  Corrado Monti, Alessandro Rozza, Giovanni Zappella, Matteo Zignani, Adam Arvidsson, and Elanor Colleoni

12:00-13:30: Lunch

13:30-15:30: Session II

- **Enhancing Sentiment Extraction from Text by Means of Arguments**
  Lucas Carstens and Francesca Toni
- **Evaluation of an Algorithm for Aspect-Based Opinion Mining Using a Lexicon-Based Approach**
  Florian Wogenstein, Johannes Drescher, Dirk Reinel, Sven Rill, and Jörg Scheidt
- **Commonsense-Based Topic Modeling**
  Dheeraj Rajagopal, Daniel Olsher, Erik Cambria, and Kenneth Kwok
- **Online Debate Summarization using Topic Directed Sentiment Analysis**
  Sarvesh Ranade, Jayant Gupta, Vasudeva Varma, and Radhika Mamidi

15:30-16:00: Coffee Break

16:00-17:30: Session III

- **RBEM: A Rule Based Approach to Polarity Detection**
  Erik Tromp and Mykola Pechenizkiy
- **Cross-lingual Polarity Detection with Machine Translation**
  Erkin Demirtas and Mykola Pechenizkiy
- **Sentribute: Image Sentiment Analysis from a Mid-level Perspective**
  Jianbo Yuan, Quanzeng You, Sean Mcdonough, and Jiebo Luo
- **Closing Remarks**

Keynote Talk

Statistical Methods for Integration and Analysis of Opinionated Text Data

Chengxiang Zhai, University of Illinois at Urbana-Champaign (USA)

Abstract

Opinionated text data such as blogs, forum posts, product reviews and online comments are increasingly available on the Web. They are very useful sources for public opinions about virtually any topics. However, because the opinions are scattered and abundant, it is a significant challenge for users to collect all the opinions about a topic and digest them efficiently. In this talk, I will present a suite of general statistical text mining methods that can help users integrate, summarize and analyze scattered online opinions to obtain actionable knowledge for decision making. Specifically, I will first present approaches to integration of scattered opinions by aligning them to a well-structured article or relevant ontology. Second, I will discuss several techniques for generating a concise opinion summary that can reveal the major sentiments and opinion points buried in large amounts of opinionated text data. Finally, I will present probabilistic general models for analyzing review data in depth to discover latent aspect ratings and relative weights placed by reviewers on different aspects. These methods are completely general and can thus help users integrate and analyze large amounts of online opinionated text data on any topic in any natural language.

About the Speaker

Chengxiang Zhai is an Associate Professor of Computer Science at the University of Illinois at Urbana-Champaign, where he is also affiliated with the Graduate School of Library and Information Science, Institute for Genomic Biology, and Department of Statistics. He received a Ph.D. in Computer Science from Nanjing University in 1990, and a Ph.D. in Language and Information Technologies from Carnegie Mellon University in 2002. He worked at Clairvoyance Corp. as a Research Scientist and a Senior Research Scientist from 1997 to 2000. His research interests include information retrieval, text mining, natural language processing, machine learning, and biomedical informatics, in which he published over 150 research papers. He is an Associate Editor of ACM Transactions on Information Systems, and Information Processing and Management, and serves on the editorial board of Information Retrieval Journal. He is a program co-chair of ACM CIKM 2004, NAACL HLT 2007, and ACM SIGIR 2009. He is an ACM Distinguished Scientist and a recipient of multiple best paper awards, Alfred P. Sloan Research Fellowship, IBM Faculty Award, HP Innovation Research Program Award, and the Presidential Early Career Award for Scientists and Engineers (PECASE).

Organizers

- Erik Cambria, National University of Singapore (Singapore)
- Bing Liu, University of Illinois at Chicago (USA)
- Yongzheng Zhang, eBay Inc. (USA)
- Yunqing Xia, Tsinghua University (China)

Program Committee

- Hisham Al-Mubaid, University of Houston at Clear Lake (USA)
- Alexandra Balahur, European Commission Joint Research Center (Italy)
- Cristina Bosco, Università di Torino (Italy)
- Ping Chen, University of Houston – Downtown (USA)
- Rossana Damiano, Università di Torino (Italy)
- Amitava Das, Norwegian University of Science and Technology (Norway)
- Dipankar Das, Jadavpur University (India)
- Giuseppe Di Fabbrizio, Amazon Inc. (USA)
- Viswanath Gopalakrishnan, Samsung Research India (India)
- Marco Grassi, Marche Polytechnic University (Italy)
- Rafael del Hoyo, Aragon Institute of Technology (Spain)
- Saif Mohammad, National Research Council (Canada)
- Muaz Niazi, Bahria University (Pakistan)
- Viviana Patti, Università di Torino (Italy)
- Rui Xia, Nanjing University of Science and Technology (China)
- Yunqing Xia, Tsinghua University (China)
- Yusheng Xie, Northwestern University (USA)
- Chengxiang Zhai, University of Illinois at Urbana-Champaign (USA)
- Lei Zhang, University of Illinois at Chicago (USA)
- Yongzheng Zhang, eBay Inc. (USA)

Acknowledgements

We would like to thank all the program committee members, contributing authors, invited speaker, and attendees for contributing to the success of the workshop. Special thanks are also extended to the SIGKDD'13 conference organizing committee for coordinating with us to put together the excellent workshop program on schedule.

Table of Contents

Identifying Purpose Behind Electoral Tweets

Saif M. Mohammad, Svetlana Kiritchenko, and Joel Martin
National Research Council Canada
Ottawa, Ontario, Canada K1A 0R6
{saif.mohammad,svetlana.kiritchenko,joel.martin}@nrc-cnrc.gc.ca

ABSTRACT

Tweets pertaining to a single event, such as a national election, can number in the hundreds of millions. Automatically analyzing them is beneficial in many downstream natural language applications such as question answering and summarization. In this paper, we propose a new task: identifying purpose behind electoral tweets—why do people post election-oriented tweets? We show that identifying purpose is related to sentiment and emotion detection, but yet significantly different. Detecting purpose has a number of applications including detecting the mood of the electorate, estimating the popularity of policies, identifying key issues of contention, and predicting the course of events. We create a large dataset of electoral tweets and annotate a few thousand tweets for purpose. We develop a system that automatically classifies electoral tweets as per their purpose, obtaining an accuracy of 44.58% on an 11-class task and an accuracy of 73.91% on a 3-class task (both accuracies well above the most-frequent-class baseline). We also show that resources developed for emotion detection are helpful for detecting purpose.

1. INTRODUCTION

The number of tweets pertaining to a single event or topic such as a national election, a natural disaster, or gun control laws, can grow to the hundreds of millions. The large number of tweets negates the possibility of a single person reading all of them to gain an overall global perspective. Thus, automatically analyzing tweets is beneficial in many downstream natural language applications such as question answering and summarization.

An important facet in understanding tweets is the question of 'Why?', that is, what is the purpose or intent of the tweet? There has been some prior work in this regard [1, 24, 33], however, they have focused on the general motivations and reasons for tweeting. For example, Naaman et al. [24] proposed the categories of: information sharing, self promotion, opinions, statements, me now, questions, presence maintenance, anecdote (me), and anecdote (others). On the other hand, the dominant reasons for tweeting vary when tweeting about specific topics and events. For example, the reasons for tweeting in national elections are very different from the reasons for tweeting during a natural disaster, such as an earthquake.

There is growing interest in analyzing political tweets in particular because of a number of applications such as determining political alignment of tweeters [13, 9], identifying contentious issues and political opinions [19], detecting the amount of polarization in the electorate [10], and so on. There is even a body of work claiming that analyzing political tweets can help predict the outcome of elections [4, 37]. However, that claim is questioned by more recent work [2].

In this paper, we propose the task of identifying the purpose behind electoral tweets. For example, some tweets are meant to criticize, some to praise, some to express disagreement, and so on. Determining the purpose behind electoral tweets can help many applications such as those listed above. There are many reasons why people criticize, praise, etc, but that is beyond the scope of this paper. For discussions on user satisfaction from tweets we refer the reader to work by Liu, Cheung, and Lee [18].

First, we automatically compile a dataset of electoral tweets using a few hand-chosen hashtags. We choose the 2012 US presidential elections as our target domain. We develop a questionnaire to annotate tweets for purpose by crowdsourcing. We analyze the annotations to determine the distributions of different kinds of purpose. We show that emotion detection alone can fail to distinguish between several different types of purpose. For example, the same emotion of dislike can be associated with many different kinds of purpose such as 'to criticize', 'to vent', and 'to ridicule'. Thus, detecting purpose provides information that is not obtainable simply by detecting sentiment or emotion.

Next, we develop a preliminary system that automatically classifies electoral tweets as per their purpose, using various features that have traditionally been used in tweet classification, such as word ngrams and emoticons, as well as features pertaining to eight basic emotions. We show that resources developed for emotion detection are also helpful for detecting purpose. We then add to this system features pertaining to hundreds of fine emotion categories. We show that these features lead to significant improvements in accuracy

WISDOM'13, August 11 2013, Chicago, USA.

above and beyond those obtained by the competitive preliminary system. The system obtains an accuracy of 44.58% on a 11-class task and an accuracy of 73.91% on a 3-class task. We publicly release all the data created as part of this project: about 1 million original tweets on the 2012 US elections, about 2,000 tweets annotated for purpose, about 1,200 tweets annotated for emotion, and the new emotion lexicon.[1]

This paper is organized as follows. We begin with related work (Section 2). We then describe how we collected and annotated the data (Sections 3.1 and 3.2). Section 3.3 gives an analysis of the annotations including distributions of various kinds of purpose, inter-annotator agreement, and confusion matrices. In Section 3.4, we flesh out the partial correlation and the distinction between purpose and affect. In Section 4, we first present a basic system to classify tweets by purpose (Section 4.1), and then we describe how we created an emotion resource pertaining to hundreds of emotions and used it to further improve performance of the basic system (Section 4.2) We present concluding remarks in Section 5.

2. RELATED WORK

There exists considerable work on tweet classification by topic [32, 17, 25]. Some of the classification work that comes close to identifying purpose is described below. Alhadi et al. [1] annotated 1000 tweets into the categories of social interaction with people, promotion or marketing, share resources, give or require feedback, broadcast alert/urgent information, require/raise funding, recruit worker, and express emotions. Naaman et al. [24] organized 3379 tweets into the categories of information sharing, self promotion, opinions, statements, me now, questions, presence maintenance, anecdote (me), and anecdote (others). Sankaranarayanan et al. [33] built a system to identify tweets pertaining to breaking news. Sriram et al. [34] annotated 5407 tweets into news, events, opinions, deals and private messages.

Tweet categorization work within a particular domain includes that by Collier, Son, and Nguyen [8], where flu-related tweets were classified into avoidance behavior, increased sanitation, seeking pharmaceutical intervention, wearing a mask, and self reported diagnosis, and work by Caragea et al. [5], where earthquake-related tweets were classified into medical emergency, people trapped, food shortage, water shortage, water sanitation, shelter needed, collapsed structure, food distribution, hospital/clinic services, and person news.

To the best of our knowledge, there is no work yet on classifying electoral or political tweets into sub-categories. As mentioned earlier, there exists work on determining political alignment of tweeters [13, 9], identifying contentious issues and political opinions [19], detecting the amount of polarization in the electorate [10], and detecting sentiment in political tweets [4, 7].

Sentiment classification of general (non-domain) tweets has received much attention [26, 14, 16]. Beyond simply positive and negative sentiment, some recent work also classifies tweets into emotions [15, 20, 31, 36]. Much of this work focused on emotions argued to be the most basic. For example, Ekman [11] proposed six basic emotions—joy, sadness, anger, fear, disgust, and surprise. Plutchik [30] argued in favor of eight—Ekman's six, trust, and anticipation. There is less work on complex emotions, such as work by Pearl and Steyvers [29] that focused on politeness, rudeness, embarrassment, formality, persuasion, deception, confidence, and disbelief.

Many of the automatic emotion classification systems use affect lexicons such as the NRC emotion lexicon [22, 23], WordNet Affect [35], and the Affective Norms for English Words.[2] Affect lexicons are lists of words and associated emotions and sentiments. We will show that affect lexicons are helpful for detecting purpose behind tweets as well.

Table 1: Query terms used to collect tweets pertaining to the 2012 US presidential elections.

#4moreyears	#Barack	#campaign2012
#dems2012	#democrats	#election
#election2012	#gop2012	#gop
#joebiden2012	#mitt2012	#Obama
#ObamaBiden2012	#PaulRyan2012	#president
#president2012	#Romney	#republicans
#RomneyRyan2012	#veep2012	#VP2012
Barack	Obama	Romney

3. DATA COLLECTION AND ANNOTATION OF PURPOSE

In the subsections below we describe how we collected tweets posted during the run up to the 2012 US presidential elections and how we annotated them for purpose by crowdsourcing.

3.1 Identifying Electoral Tweets

We created a corpus of tweets by polling the Twitter Search API, during August and September 2012, for tweets that contained commonly known hashtags pertaining to the 2012 US presidential elections. Table 1 shows the query terms we used. Apart from 21 hashtags, we also collected tweets with the words Obama, Barack, or Romney. We used these additional terms because they were the names of the two presidential candidates. Further, the probability that these words were used to refer to someone other than the presidential candidates was low.

The Twitter Search API was polled every four hours to obtain new tweets that matched the query. Close to one million tweets were collected, which we will make freely available to the research community.[3] The query terms which produced the highest number of tweets were those involving the names of the presidential candidates, as well as #election2012, #campaign, #gop, and #president.

[1]Email Saif Mohammad: saif.mohammad@nrc-cnrc.gc.ca.

[2]http://csea.phhp.ufl.edu/media/anewmessage.html

[3]Note that Twitter imposes restrictions on direct distribution of tweets, but allows the distribution of tweet ids. One may download tweets using tweet ids and third party tools, provided those tweets have not been deleted by the people who posted them.

We used the metadata tag "iso_language_code" to identify English tweets. Since this tag does not always correctly reflect the language of the tweet, we also discarded tweets that did not have at least two valid English words. We used the Roget Thesaurus as the English word inventory. This step also helps discard very short tweets and tweets with a large proportion of misspelled words.

Since we were interested in determining the purpose behind the tweets, we decided to focus on original tweets as opposed to retweets. Retweets can easily be identified through the presence of RT, rt, or Rt in the tweet (usually in the beginning of the post). All such tweets were discarded.

3.2 Annotating Purpose by Crowdsourcing

We used Amazon's Mechanical Turk service to crowdsource the annotation of the electoral tweets.[4] We randomly selected about 2,000 tweets, each by a different Twitter user. We asked a series of questions for each tweet. Below is the questionnaire for an example tweet:

Purpose behind US election tweets

Tweet: Mitt Romney is arrogant as hell.

Q1. Which of the following best describes the purpose of this tweet?

- to point out hypocrisy or inconsistency
- to point out mistake or blunder
- to disagree
- to ridicule
- to criticize, but none of the above
- to vent

- to agree
- to praise, admire, or appreciate
- to support

- to provide information without emotion
- none of the above

Q2. Is this tweet about US politics and elections?

- Yes, this tweet is about US politics and elections.
- No, this tweet has nothing to do with US politics or anybody involved in it.

These questionnaires are called *HITs (human intelligence tasks)* in Mechanical Turk parlance. We posted 2042 HITs corresponding to 2042 tweets. We requested responses from at least three annotators for each HIT. The response to a HIT by an annotator is called an *assignment*. In Mechanical Turk, an annotator may provide assignments for as many HITs as they wish. Thus, even though only three annotations are requested per HIT, about 400 annotators contribute assignments for the 2,042 tweets. The number of assignments completed by the annotators followed a zipfian distribution.

Even though it is possible that more than one option may apply for a tweet, we allowed the Turkers to select only one option for each question. We did this to encourage annotators to select the option that best answers the questions. We wanted to avoid situations where an annotator selects multiple options just because they are vaguely relevant to the question.

[4]https://www.mturk.com/mturk/welcome

Table 2: The histogram of the number of annotations of tweets. 'annotns' is short for annotations.

annotns/tweet	# of tweets	# of annotns
1	181	181
2	594	1188
3	1121	3363
4	60	240
≥5	88	1509
all	2042	6481

We created an initial set of categories of purpose by consultations with colleagues and analysis of a small set of tweets. We further refined the set of categories after a pilot annotation project by removing categories that were not represented in the data and also categories that were confused with others. For example, we removed the category 'to entertain' as it was found to intersect with several other categories.

Observe that we implicitly grouped the final options for Q1 into three coarse categories by putting extra vertical space between the groups. These coarse categories correspond to *oppose* (to point out hypocrisy, to point out mistake, to disagree, to ridicule, to criticize, to vent), *favour* (to agree, to praise, to support), and *other*. Even though there is some redundancy among the fine categories, they are more precise and may help annotation. Eventually, however, it may be beneficial to combine two or more categories for the purposes of automatic classification. The amount of combining will depend on the task at hand, and can be done to the extent that anywhere from eleven to two categories remain.

3.3 Annotation Analyses

The Mechanical Turk annotations were done over a period of one week. For each annotator, and for each question, we calculated the probability with which the annotator agrees with the response chosen by the majority of the annotators. We identified poor annotators as those that had an agreement probability that was more than two standard deviations away from the mean. All annotations by these annotators were discarded. Table 2 gives a histogram of the number of annotations of the remaining tweets. There were 1121 tweets with exactly three annotations.

We determined whether a tweet is to be assigned a particular category based on strong majority. That is, a tweet belongs to category X if it is annotated with X more often than all other categories combined. Percentage of tweets in each of the 11 categories of Q1 are shown in Table 3. Observe that the majority category for purpose is 'to support'—26.49% of the tweets were identified as having the purpose 'to support'. Table 4 gives the distributions of the three coarse categories of purpose. Observe, that the political tweets express disagreement (58.07%) much more than support (31.76%).

Table 5 gives the distributions for question 2. Observe that a large majority (95.56%) of the tweets are relevant to US politics and elections. This shows that the hashtags shown earlier in Table 1 are effective in identifying political tweets.

Table 3: Percentage of tweets in each of the eleven categories of Q1. Only those tweets that were annotated by at least two annotators were included. A tweet belongs to category X if it is annotated with X more often than all other categories combined. There were 1072 such tweets in total.

Purpose of tweet	Percentage of tweets
favour	
to agree	0.47
to praise, admire, or appreciate	15.02
to support	**26.49**
oppose	
to point out hypocrisy or inconsistency	7.00
to point out mistake or blunder	3.45
to disagree	2.52
to ridicule	15.39
to criticize, but none of the above	7.09
to vent	8.21
other	
to provide information without any emotional content	13.34
none of the above	1.03
all	100.0

Table 4: Percentage of tweets in each of the three coarse categories of Q1. Only those tweets that were annotated by at least two annotators were included. A tweet belongs to category X if it is annotated with X more often than all other categories combined. There were 1672 such tweets in total. The annotator agreement on the three categories is larger than on eleven categories.

Category	Percentage of tweets
oppose	**58.07**
favour	31.76
other	10.17
all	100.0

3.3.1 Inter-Annotator Agreement

We calculated agreement on the full set of annotations, and not just on the annotations with a strong majority as described in the previous section. One way to gauge the amount of agreement among annotators is to examine the number of times all three annotators agree (majority class size = 3), the number of times two out of three annotators agree (majority class size = 2), and the number of times all three annotators choose different options (majority class size = 1).

Table 6 gives the distributions of the majority classes. Higher numbers for the larger class sizes indicate higher agreement. For example, for 22.4% of the tweets all three annotators gave the same answer for question 1 (Q1). The agreement is much higher if one only considers the coarse categories of 'oppose', 'favour', and 'other'—these numbers are shown in the row marked Q1'. The agreement for question 2 was substantially high. This was expected as it is a relatively straightforward question. The numbers in the table are calculated from tweets with exactly three annotations.

Table 5: Percentage of tweets in each of the two categories of Q2.

Relevance	Percentage of tweets
pertaining to US politics and elections	**95.56**
not pertaining to US politics and elections	4.44
all	100.0

Table 6: Percentage of tweets having majority class size (MCS) of 1, 2, and 3. Note: Q is short for question.

	MCS-1	MCS-2	MCS-3
Q1	29.5	48.1	22.4
Q1'	2.2	31.7	66.1
Q2	0.0	5.7	94.3

Table 7 shows *inter-annotator agreement (IAA)*, for the two questions—the average percentage of times two annotators agree with each other. IAA gives us an understanding of the degree of agreement through a single number. Observe that the agreement is only moderate for the eleven fine categories of purpose (43.58%), but much higher when considering the coarser categories (83.81%).

Another way to gauge agreement is by calculating the average probability with which an annotator picks the majority class. Consider the example below: Each tweet is annotated by 3 different annotators. X annotates 10 tweets. Six of the times, X's answer for Q1 is the answer that has a majority (in case of 3 annotators, this means that at least one other annotator also gave the same answer as X for 6 of the 10 tweets). Thus the probability with which X picks the majority class is 6/10. The last column in Table 7 shows the *average probability of picking the majority class (APMS)* by the annotators (higher numbers indicate higher agreement). Overall, we observe that there is strong agreement between annotators at identifying whether the purpose of a tweet is to oppose, to favour, or something else.

3.3.2 Confusion Matrix

Human annotators may disagree with each other because two or more options may seem appropriate for a given tweet. There also exist tweets where the purpose is unclear. Table 8 shows the confusion matrix for question 1. The rows and columns of the matrix correspond to the eleven options. The value in a particular cell, say for row x and column y, is the number of annotations that were assigned label y even though the majority votes for each of those tweets were for x. The highest number in each row is shown in bold. The cells in the diagonal correspond to the number of instances for which the annotations matched the majority vote. For high agreement, one would want higher numbers in the diagonal, which is what we observe in Table 8.

We can identify options that tend to be confused for each other by noting non-diagonal cells with high values. For example, consider cell r7–c8. The relatively large number indicates that 'to ridicule' is sometimes confused with 'to

Table 8: Confusion Matrix: Question 1 (fine-grained). The value in a particular cell, say for row x and column y, is the number of annotations that were assigned label y even though the majority votes for each of those tweets were for x. The highest number in each row is shown in bold.

		c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	c11
favour												
to agree:	r1	**20**	5	9	2	1	2	0	3	0	4	0
to praise, admire, or appreciate:	r2	0	**291**	61	1	1	5	1	5	4	3	0
to support:	r3	1	43	**565**	5	4	23	7	18	5	22	3
oppose												
to point out hypocrisy or inconsistency:	r4	2	2	14	**123**	15	26	10	64	11	5	0
to point out mistake or blunder:	r5	0	6	16	6	**84**	29	15	46	1	3	0
to disagree:	r6	0	0	5	10	2	**145**	10	5	5	1	0
to ridicule:	r7	3	11	28	9	16	37	**274**	60	15	4	0
to criticize, but none of the above:	r8	1	0	22	8	5	49	30	**227**	9	3	0
to vent:	r9	7	12	35	5	11	37	22	45	**155**	7	1
other												
to provide information without any emotional content:	r10	2	11	39	1	4	8	11	19	8	**259**	4
none of the above:	r11	3	6	10	1	4	5	7	3	6	10	**19**

Table 7: Agreement statistics: inter-annotator agreement (IAA) and average probability of choosing the majority class (APMS).

	IAA	APMS
Q1	43.58	0.520
Q1'	83.81	0.855
Q2	96.76	0.974

criticize, but none of the above'. Similarly, we find that 'to point out hypocrisy or inconsistency' and 'to point out mistake or blunder' can also be confused with 'to criticize, but none of the above' (r4–c8 and r5–c8). Note however, that the labels are not confused as strongly in the other direction. For example, tweets that have a purpose of 'to criticize' are not confused as much with 'to point out hypocrisy' (r8–c4). This suggests that the category 'to criticize, but none of the above' serves as a hold-back for other finer-grained categories of 'oppose' and, therefore, is often chosen by annotators for less clear messages. A similar situation occurs in the 'favour' group, where the confusion occurs mostly between a more general category 'to support' and more specific categories 'to agree' and 'to praise, admire, or appreciate'.

Note that in a particular application, one may choose only a subset of the eleven categories that are most relevant. For example, one may combine 'to point out hypocrisy', 'to point out mistake', and 'to criticize, but none of the above' into a single category, and distinguish it from other oppose categories such as 'to disagree' and 'to ridicule'.

Table 9 shows the confusion matrix within the coarse categories of question 1. The confusion between the coarse categories is lower than among the finer categories, but yet there exist instances when 'favour' is confused with 'oppose', and vice versa. Table 10 shows the confusion matrix for question 2. Only a very small number of instances are confused with the wrong option for this question.

Table 9: Confusion Matrix: Question 1' (coarse grained).

		c1	c2	c3
favour:	r1	**941**	136	37
oppose:	r2	75	**1705**	29
other:	r3	40	88	**312**

Table 10: Confusion Matrix: Question 2.

		c1	c2
not pertaining to US politics and elections:	r1	**106**	38
pertaining to US politics and elections:	r2	26	**3193**

3.4 Distinctions between purpose and affect

The task of detecting purpose is related to sentiment and emotion classification. Intuitively, the three broad categories of purpose, 'oppose', 'favour', and 'other', roughly correspond to negative, positive, and objective sentiment. Also, some fine-grained categories seem to partially correlate with emotions. For example, when angry, a person vents. When overcome with admiration, a person praises the object of admiration.

To further investigate the relation between purpose and emotion, we annotated a portion of the tweets by crowdsourcing with one of 19 emotions: acceptance, admiration, amazement, anger, anticipation, calmness, disappointment, disgust, dislike, fear, hate, indifference, joy, like, sadness, surprise, trust, uncertainty, and vigilance. Similar to the annotation of purpose, each tweet was annotated by at least two judges, and tweets with no strong majority were discarded.

Table 11 shows the percentage of tweets pertaining to different emotions. Only high-frequency categories of purpose and emotion are shown. As expected, the tweets with the purpose 'favour' mainly convey the emotions of admiration,

Table 11: Percentage of different purpose tweets pertaining to different emotions. Low-frequency categories of purpose and emotion are omitted. The highest number for each category of purpose is shown in bold.

	admiration	anticipation	joy	dislike	disappointment	disgust	anger
favour							
to praise, admire, or appreciate	**67**	4	25				
to support	**33**	21	21	4	2		7
oppose							
to point out hypocrisy or inconsistency				**61**		17	11
to point out mistake or blunder				**77**		15	8
to disagree			14	**43**		14	29
to ridicule			7	**66**		7	18
to criticize, but none of the above				**47**	11	16	16
to vent			4	24	12	8	**36**

anticipation, and joy. On the other hand, the tweets with the purpose 'oppose' are mostly associated with negative emotions such as dislike, anger, and disgust. The purpose 'to praise, admire, or appreciate' is highly correlated with the emotion admiration.

Note that most of the tweets with the purpose 'to point out hypocrisy', 'to point out mistake', 'to disagree', 'to ridicule', 'to criticize', and even many instances of 'to vent' are associated with the emotion dislike. Thus, a system that only determines emotion and not purpose will fail to distinguish between these different categories of purpose. It is possible for people to have the same emotion of dislike and react differently: either by just disagreeing, pointing out the mistake, criticizing, or resorting to ridicule.

4. DETECTING PURPOSE

In this section, we investigate the usefulness of emotion resources in automatically detecting purpose. We train an automatic classifier over an extensive set of features drawn from those used for sentiment analysis of social media texts [27, 3, 21] as well as emotion features and determine the impact of each feature group on classifier performance.

We used a Support Vector Machine (SVM) classifier as they have been shown to be effective on text categorization tasks and robust on large feature spaces. We used the LibSVM package [6] with linear kernel. Parameter C was chosen by cross-validation on the training portion of the data (i.e., the nine training folds). We first classified the tweets into one of eleven categories of purpose. In a second set of experiments, the eleven fine-grained categories were combined into 3 coarse-grained - 'oppose', 'favour', and 'other' - as was described earlier. In each experiment, ten-fold stratified cross-validation was repeated ten times, and the results were averaged. Paired t-test was used to confirm the significance of the results.

The gold labels were determined by strong majority voting. Tweets with less than 2 annotations or with no majority labels were discarded. Thus, the dataset consisted of 1072 tweets for the 11-category task, and 1672 tweets for the 3-category task. The tweets were normalized by replacing URLs with http://someurl and userids with @someuser. The tweets were tokenized and tagged with parts of speech using the Carnegie Mellon University Twitter NLP tool [12].

4.1 A Basic System for Purpose Classification

We employed commonly used text classification features such as ngrams, part-of-speech, and punctuations, as well as common Twitter-specific features such as emoticons and hashtags. Additionally, we hypothesized that the purpose of tweets is guided by the emotions of the tweeter. Thus we explored certain emotion features as well. Each tweet was represented with the following groups of features:

- n-grams: presence of n-grams (contiguous sequences of 1, 2, 3, and 4 tokens), skipped n-grams (n-grams with one token replaced by *), character n-grams (contiguous sequences of 3, 4, and 5 characters);
- POS: number of occurrences for each part-of-speech tag;
- word clusters: presence of words from each of the 1000 word clusters provided by the Twitter NLP tool [12]. These clusters were produced with the Brown clustering algorithm on 56 million English-language tweets. They serve as alternative representation of tweet content, reducing the sparcity of the token space.
- all-caps: number of words with all characters in upper case;
- NRC Emotion Lexicon: We used the NRC Emotion Lexicon [22] to incorporate affect features. The lexicon consists of 14,182 words manually annotated with 8 basic emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust) and 2 polarities (positive, negative). Each word can have zero, one, or more associated emotions and zero or one polarity. For each tweet we counted:
  - number of words associated with each emotion
  - number of nouns, verbs, etc., associated with each emotion
  - number of all-caps words associated with each emotion
  - number of hashtags associated with each emotion
- negation: the number of negated contexts. Following [28], we defined a negated context as a segment of a tweet that starts with a negation word (e.g., 'no', 'shouldn't') and ends with one of the punctuation marks: ',', '.', ':', ';', '!', '?'. A negated context affects the n-gram and Emotion Lexicon features: each word and associated with it emotion in a negated context become

Table 12: Accuracy of the automatic classification on 11-category and 3-category problems. The lower bound is the percentage of the majority class.

	11-class	3-class
majority class	26.49	58.07
SVM	43.56	73.91

Table 13: Per category precision (P), recall (R), and F1 score of the classification on the 11-category problem. Micro-averaged P, R, and F1 are equal to accuracy since the categories are mutually exclusive.

category	# inst.	P	R	F1
favour				
to agree	5	0	0	0
to praise	161	57.59	50.43	53.77
to support	284	49.35	69.47	57.71
oppose				
to point out hypocrisy	75	30.81	21.2	25.12
to point out mistake	37	0	0	0
to disagree	27	0	0	0
to ridicule	165	31.56	43.76	36.67
to criticize	76	22.87	9.87	13.79
to vent	88	36.06	23.07	28.14
other				
to provide information	143	45.14	50.63	47.73
none of the above	11	0	0	0
micro-ave		43.56	43.56	43.56

negated (e.g., 'not perfect' becomes 'not perfect_NEG', 'EMOTION_trust' becomes 'EMOTION_trust_NEG'). The list of negation words was adopted from Christopher Potts' sentiment tutorial.[5]

- punctuation: the number of contiguous sequences of exclamation marks, question marks, and both exclamation and question marks;
- emoticons: presence/absence of positive and negative emoticons. The polarity of an emoticon was determined with a simple regular expression adopted from Christopher Potts' tokenizing script.[6]
- hashtags: the number of hashtags;
- elongated words: the number of words with one character repeated more than 2 times, e.g. 'soooo'.

Table 12 presents the results of the automatic classification for the 11-category and 3-category problems. For comparison, we also provide the accuracy of a simple baseline classifier that always predicts the majority class.

Table 13 shows the classification results broken-down by category. As expected, the categories with larger amounts of labeled examples ('to praise', 'to support', 'to provide information') have higher results. However, for one of the higher frequency categories, 'to ridicule', the F1-score is relatively low. This category incorporates irony, sarcasm, and humour, the concepts that are hard to recognize, especially in a very restricted context of 140 characters. The four low-frequency categories ('to agree', 'to point out mistake or blunder', 'to disagree', 'none of the above') did not have enough training data for the classifier to build adequate models. The categories within 'oppose' are more difficult to distinguish among than the categories within 'favour'. However, for the most part this can be explained by the larger number of categories (6 in 'oppose' vs. 3 in 'favour') and, consequently, smaller sizes of the individual categories.

[5]http://sentiment.christopherpotts.net/lingstruc.html
[6]http://sentiment.christopherpotts.net/tokenizing.html

Table 14: Accuracy of classification with one of the feature groups removed. Numbers in bold represent statistically significant difference with the accuracy of the 'all features' classifier (first line) with 95% confidence.

Experiment	11-class	3-class
all features	43.56	73.91
all - n-grams	**39.51**	**71.02**
all - NRC emotion lexicon	**42.27**	**72.21**
all - parts of speech	**42.63**	**73.55**
all - word clusters	**43.24**	**73.24**
all - negation	**43.18**	**73.36**
all - (all-caps, punctuation, emoticons, hashtags)	43.38	73.87

Table 15: Accuracy of classification using different lexicons on the 11-class problem. Numbers in bold represent statistically significant difference with the accuracy of the classifier using the NRC Emotion Lexicon (first line) with 95% confidence.

Lexicon	Accuracy
NRC Emotion Lexicon	43.56
Hashtag Lexicon	44.35
both lexicons	**44.58**

We investigated the usefulness of each feature group by repeating the above classification process and each time removing one of the feature groups. Table 14 shows the results of these ablation experiments for the 11-category and 3-category problems. In both cases, the emotion lexicon features were found to be helpful and provided significant gains, second only to the ngram features.

4.2 Adding features pertaining to hundreds of fine emotions

Since the emotion lexicon had a significant impact on the results, we further created a wide-coverage twitter-specific lexical resource following on work by Mohammad [20]. [20] showed that emotion-word hashtagged tweets are a good source of labeled data for automatic emotion processing. Those experiments were conducted using tweets pertaining to the six Ekman emotions because labeled evaluation data

exists for only those emotions. However, a significant advantage of using hashtagged tweets is that we can collect large amounts of labeled data for any emotion that is used as a hashtag by tweeters. Thus we polled the Twitter API and collected a large corpus of tweets pertaining to a few hundred emotions.

We used a list of 585 emotion words compiled by Zeno G. Swijtink as the hashtagged query words.[7] Note that we chose not to dwell on the question of whether each of the words in this set is truly an emotion or not. Our goal was to create and distribute a large set of affect-labeled data, and users are free to choose a subset of the data that is relevant to their application. We calculated the pointwise mutual information (PMI) between an emotional hashtag and a word appearing in tweets. The PMI represents a degree of correlation between the word and emotion, with larger scores representing stronger correlations. Consequently, the pairs (word, hashtag) that had positive PMI were pulled together into a new word–emotion association resource, that we call *Hashtag Emotion Lexicon*. The lexicon contains around 10,000 words with associations to 585 emotion-word hashtags.

We used the Hashtag Lexicon for classification by creating features in the same way as we did for the NRC Emotion Lexicon. Since the Hashtag Lexicon additionally provides real-valued scores of association, for each tweet, we calculated the sum of these scores instead of simply counting the number of emotion-associated words. Table 15 shows the results. The Hashtag Lexicon improved the performance of the classifier on the 11-category task. Even better results were obtained when both lexicons were employed (the improvement over the NRC Emotion Lexicon is statistically significant)[8].

5. CONCLUSIONS

Tweets are playing a growing role in the public discourse on politics. In this paper, we explored the purpose behind such tweets. Detecting purpose has a number of applications including detecting the mood of the electorate, estimating the popularity of policies, identifying key issues of contention, and predicting the course of events. We compiled a dataset of 1 million tweets pertaining to the 2012 US presidential elections using relevant hashtags. We designed an online questionnaire and annotated a few thousand tweets for purpose via crowdsourcing. We analyzed these tweets and showed that a large majority convey emotional attitude towards someone or something. Further, the number of messages posted to oppose someone or something were almost twice the number of messages posted to offer support.

We developed a classifier to automatically classify electoral tweets as per their purpose. It obtained an accuracy of 44.58% on a 11-class task and an accuracy of 73.91% on a 3-class task (both accuracies well above the most-frequent-class baseline). We found that word–emotion association resources such as the NRC Emotion Lexicon and the Hashtag Emotion Lexicon are helpful for detecting purpose. However, we also showed that emotion detection alone can fail to distinguish between several kinds of purpose. We make all the data created as part of this research freely available.

In this paper, we relied only on the target tweet as context. However, it might be possible to further improve results by modeling user behaviour based on multiple past tweets. We are also interested in using purpose-annotated tweets as input in a system that automatically summarizes political tweets. Finally, we hope that a better understanding of purpose of tweets will help drive the political discourse towards issues and concerns most relevant to the people.

[7] http://www.sonoma.edu/users/s/swijtink/teaching/philosophy_101/paper1/listemotions.htm

[8] Using the Hashtag Lexicon on the 3-category task did not show any improvement. This is probably because in the 3-category task the information about positive and negative sentiment provides the most gain.

6. REFERENCES

[1] A. C. Alhadi, S. Staab, and T. Gottron. Exploring User Purpose Writing Single Tweets. In *WebSci'11: Proceedings of the 3rd International Conference on Web Science*, 2011.

[2] D. G. Avello. "I Wanted to Predict Elections with Twitter and all I got was this Lousy Paper" – A Balanced Survey on Election Prediction using Twitter Data. *arXiv*, 1204.6441, 2012.

[3] L. Barbosa and J. Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of Coling: Poster Volume*, pages 36–44, Beijing, China, August 2010.

[4] A. Bermingham and A. Smeaton. On Using Twitter to Monitor Political Sentiment and Predict Election Results. In *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011)*, pages 2–10, Chiang Mai, Thailand, 2011. Asian Federation of Natural Language Processing.

[5] C. Caragea, M. McNeese, A. Jaiswal, G. Traylor, H. Kim, P. Mitra, D. Wu, A. Tapia, C. Giles, J. Jansen, and J. Yen. Classifying Text Messages for the Haiti Earthquake. In *Proceedings of the 8th International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, Lisbon, Portugal, 2011.

[6] C.-C. Chang and C.-J. Lin. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] J. E. Chung and E. Mustafaraj. Can Collective Sentiment Expressed on Twitter Predict Political Elections? In W. Burgard and D. Roth, editors, *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, California, USA, 2011. AAAI Press.

[8] N. Collier, N. Son, and N. Nguyen. OMG U got flu? Analysis of Shared Health Messages for Bio-surveillance. *Journal of Biomedical Semantics*, 2(Suppl 5):S9, 2011.

[9] M. D. Conover, B. Goncalves, J. Ratkiewicz, A. Flammini, and F. Menczer. Predicting the Political Alignment of Twitter Users. In *IEEE Third International Conference on Privacy Security Risk and Trust and IEEE Third International Conference on Social Computing*, pages 192–199. IEEE, 2011.

[10] M. D. Conover, J. Ratkiewicz, M. Francisco, B. Gonc, A. Flammini, and F. Menczer. Political Polarization

on Twitter. *Networks*, 133(26):89–96, 2011.

[11] P. Ekman. An Argument for Basic Emotions. *Cognition and Emotion*, 6(3):169–200, 1992.

[12] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith. Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2011.

[13] J. Golbeck and D. Hansen. Computing Political Preference Among Twitter Followers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1105–1108, New York, NY, 2011. ACM.

[14] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao. Target-Dependent Twitter Sentiment Classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-2011)*, pages 151–160, 2011.

[15] S. Kim, J. Bak, and A. H. Oh. Do You Feel What I Feel? Social Aspects of Emotions in Twitter Conversations. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2012.

[16] E. Kouloumpis, T. Wilson, and J. Moore. Twitter Sentiment Analysis: The Good the Bad and the OMG! In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[17] K. Lee, D. Palsetia, R. Narayanan, M. M. A. Patwary, A. Agrawal, and A. Choudhary. Twitter Trending Topic Classification. In *Proceedings of the IEEE 11th International Conference on Data Mining Workshops (ICDMW)*, pages 251–258. IEEE, 2011.

[18] I. L. B. Liu, C. M. K. Cheung, and M. K. O. Lee. *Understanding Twitter Usage: What Drive People Continue to Tweet*, pages 928–939. 2010.

[19] D. Maynard and A. Funk. Automatic Detection of Political Opinions in Tweets. In *The Semantic Web: ESWC 2011 Workshops*, pages 88–99. Springer, 2011.

[20] S. Mohammad. #Emotional Tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 246–255, Montréal, Canada, 2012. Association for Computational Linguistics.

[21] S. M. Mohammad, S. Kiritchenko, and X. Zhu. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA, June 2013.

[22] S. M. Mohammad and P. D. Turney. Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon. In *Proceedings of the NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, LA, California, 2010.

[23] S. M. Mohammad and P. D. Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 2013.

[24] M. Naaman, J. Boase, and C.-H. Lai. Is It Really About Me?: Message Content in Social Awareness Streams. In *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work*, CSCW '10, pages 189–192, New York, NY, 2010. ACM.

[25] K. Nishida, R. Banno, K. Fujimura, and T. Hoshide. Tweet Classification by Data Compression. In *Proceedings of the International Workshop on Detecting and Exploiting Cultural Diversity on the Social Web*, pages 29–34. ACM, 2011.

[26] A. Pak and P. Paroubek. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *Proceedings of LREC*, 2010.

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

[28] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: Sentiment Classification Using Machine Learning Techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 79–86, Philadelphia, PA, 2002.

[29] L. Pearl and M. Steyvers. Identifying Emotions, Intentions, and Attitudes in Text Using a Game with a Purpose. In *Proceedings of the NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, Los Angeles, California, 2010.

[30] R. Plutchik. A General Psychoevolutionary Theory of Emotion. *Emotion: Theory, research, and experience*, 1(3):3–33, 1980.

[31] D. Quercia, J. Ellis, L. Capra, and J. Crowcroft. Tracking "Gross Community Happiness" from Tweets. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, CSCW '12, pages 965–968, New York, NY, 2012. ACM.

[32] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. In *Proceedings of the 19th International Conference on World Wide Web*, pages 851–860. ACM, 2010.

[33] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling. TwitterStand: News in Tweets. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, pages 42–51, New York, NY, 2009. ACM.

[34] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short Text Classification in Twitter to Improve Information Filtering. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 841–842, New York, NY, 2010. ACM.

[35] C. Strapparava and A. Valitutti. WordNet-Affect: An Affective Extension of WordNet. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC-2004)*, pages 1083–1086, Lisbon, Portugal, 2004.

[36] K. Tsagkalidou, V. Koutsonikola, A. Vakali, and K. Kafetsios. Emotional Aware Clustering on Micro-blogging Sources. In *Proceedings of the Conference on Affective Computing and Intelligent Interaction*, pages 387–396, Memphis, TN, 2011.

[37] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Election Forecasts With Twitter: How 140 Characters Reflect the Political Landscape. *Social Science Computer Review*, 29(4):402–418, 2010.

Combining Strengths, Emotions and Polarities for Boosting Twitter Sentiment Analysis

Felipe Bravo-Marquez
PRISMA Research Group
Department of Computer Science, University of Chile, Chile
Yahoo! Labs Santiago, Chile
fbravo@dcc.uchile.cl

Marcelo Mendoza
Universidad Técnica Federico Santa María, Chile
Yahoo! Labs Santiago, Chile
marcelo.mendoza@usm.cl

Barbara Poblete
PRISMA Research Group
Department of Computer Science, University of Chile, Chile
Yahoo! Labs Santiago, Chile
bpoblete@dcc.uchile.cl

ABSTRACT

Twitter sentiment analysis or the task of automatically retrieving opinions from tweets has received an increasing interest from the web mining community. This is due to its importance in a wide range of fields such as business and politics. People express sentiments about specific topics or entities with different strengths and intensities, where these sentiments are strongly related to their personal feelings and emotions. A number of methods and lexical resources have been proposed to analyze sentiment from natural language texts, addressing different opinion dimensions. In this article, we propose an approach for boosting Twitter sentiment classification using different sentiment dimensions as meta-level features. We combine aspects such as opinion strength, emotion and polarity indicators, generated by existing sentiment analysis methods and resources. Our research shows that the combination of sentiment dimensions provides significant improvement in Twitter sentiment classification tasks such as polarity and subjectivity.

Categories and Subject Descriptors

I.2.7.7 [**Artificial Intelligence**]: Natural Language Processing—*Text Analysis*

General Terms

Experimentation, Measurement

Keywords

Sentiment Classification, Twitter, Meta-level features

1. INTRODUCTION

Inherent in human nature is the need to express particular points of view and feelings about specific topics or entities. Opinions reveal beliefs about specific matters commonly considered to be subjective.

Social media has opened new possibilities for people to interact. Microblogging platforms allow real-time sharing of comments and opinions. Twitter, which has become the most popular microblogging platform, has millions of users that spread millions of personal posts on a daily basis. The rich and enormous volume of data propagated through social media offers enormous opportunities for the study of social human subjectivity.

Manual classification of millions of posts for opinion mining tasks is an unfeasible effort at human scale. Several methods have been proposed to automatically infer human opinions from natural language texts. Due to the inherent subjectivity of the data, this problem is still an open problem in the field.

Opinions are multidimensional semantic artifacts. When people are exposed to information regarding a topic or entity, they normally respond to this external stimuli by developing a personal point of view or orientation. This orientation reveals how the opinion holder is polarized by the entity. Additionally, people manifest emotions through opinions, which are the driving forces behind motivations and personal dispositions. That means that emotions and polarities are mutually influenced by each other, conditioning opinion intensities and emotional strengths.

Computational sentiment analysis methods attempt to measure different opinion dimensions. A number of methods for polarity estimation have been proposed in [3, 6, 7, 16] discussed in depth in Section 2. By transforming polarity estimation into a classification problem with three polarity classes -positive, negative and neutral- supervised and unsupervised approaches have been explored to fulfill this task. In the case of the unsupervised approaches, a number of lexicon resources with with positive and negative scores for words have been released. Another related task is the detection of *subjectivity*, which is the specific task of separating factual from opinionated text. This problem has also been addressed by using supervised approaches [25]. Opinion intensities (strengths) have also been measured. From a strength scored method, SentiStrength [23] can estimate positive and negative strength scores at sentence level. Finally, emotion estimation has also been addressed by developing lexicons. The Plutchik's wheel of emotions was proposed in [21]. The wheel is composed by four pairs of opposite emotion states: **joy-trust**, **sadness-anger**, **surprise-fear**, and **anticipation-disgust**. Mohammad et.al [14] labeled a number of words according to Plutchik emotional categories, developing the NRC word-emotion association lexicon.

According to the previous paragraphs, we can see that sentiment analysis tools focus on different scopes within opinions. Although these scopes are very difficult to categorize explicitly, we propose the following categories:

1. **Polarity**: These methods and resources aim towards extracting polarity information from a passage. Polarity-oriented

WISDOM'13, August 11 2013, Chicago, USA.

methods normally return a categorical variable whose possible values are positive, negative and neutral. On the other hand, polarity-oriented lexical resources are composed by lists of positive and negative words.

2. **Emotion**: Methods and resources focused on extracting emotion or mood states from a text passage. An emotion-oriented method should classify the message to an emotional category such as sadness, joy, surprise, among others. An emotion-oriented lexical resources should provide a list of words or expressions marked according to different emotion states.

3. **Strength**: These methods and resources provide intensity levels according to a certain sentiment dimension which can have a polarity or an emotional scope. Strength-oriented methods return different numerical scores indicating the intensity or the strength of an opinion dimension expressed in a text passage. For instance, numerical scores indicating the level of positivity, negativity or another emotional dimension. Strength-oriented lexical resources provide lists of opinion words together with intensity scores regarding an opinion dimension.

In this article we propose to efficiently combine existing sentiment analysis methods and resources focused the main scopes discussed above. Our goal is to improve two major sentiment analysis tasks: 1) Subjectivity classification, and 2) Polarity classification. We combine all of these aspects as input features in a sentiment classifier using supervised learning algorithms. To validate our approach we evaluate our classifiers on two existing datasets. Our results show that the composition of these features achieves significant improvements over single approaches. This, indicates that strength, emotion and polarity-based resources are complementary, addressing different dimensions of the same problem. Therefore, a tandem approach should be more appropriate.

To the best of our knowledge, this is the first study to combine polarity, emotion, and strength oriented sentiment analysis lexical resources with existing opinion mining methods as meta-level features for boosting sentiment classification performance.

This article is organized as follows. In Section 2 we provide a review of existing lexical resources and discuss related work on Twitter sentiment analysis. In Section 4.4 we describe our approach for Twitter sentiment classification as well as the features that are used in our classification scheme. The experimental results are presented in Section 4. Finally, we conclude in Section 5 with a brief discussion.

2. RELATED WORK

2.1 Lexical Resources for Sentiment Analysis

The development of lexical resources for sentiment analysis has gathered attention from the computational linguistic community. Wilson et al. [25] labeled a list of English words in positive and negative categories, releasing the Opinion Finder lexicon. Bradley and Lang [3] released ANEW, a lexicon with affective norms for English words. The application of ANEW to Twitter was explored by Nielsen [16], leveraging the AFINN lexicon. Esuli and Sebastiani [6] and later Baccianella et al. [1] extended the well known Wordnet lexical database [13] by introducing sentiment ratings to a number of synsets, creating SentiWordnet. The development of lexicon resources for strength estimation was addressed by Thelwall et al. [23], leveraging SentiStrength. Finally, NRC, a lexicon resource for emotion estimation was recently released by Mohammad and Turney [14], where a number of English words were tagged with emotion ratings, according to the emotional wheel taxonomy introduced by Plutchik [21].

Besides the syntactic-level resources for sentiment analysis presented above, other type of resources have been elaborated for a semantic-level analysis refered as concept-based. Concept-based approaches conduct a semantic analysis of the text using semantic knowledge bases such as web ontologies [17] and semantic networks [18]. In this manner, concept-based methods allow the detection of subjective information which can be expressed implicitly in a text passage. A publicly available concept-based resource to extract sentiment information from common sense concepts is *SenticNet 2*[1]. This resource was built using both graph-mining and dimensionality-reduction techniques [4].

2.2 Twitter Sentiment Analysis

Twitter users tend to post opinions about products or services [19]. **Tweets** (user posts on Twitter) are short and usually straight to the point messages. Therefore, tweets are considered as an interesting resource for sentiment analysis. Common tasks of opinion mining that can be applied to Twitter data are sentiment classification and opinion identification. As Twitter messages are at most, 140-characters long, a sentence-level classification approach can be adopted, assuming that tweets express opinions about one single entity. Furthermore, retrieving messages from Twitter is a straightforward task, through the use of the Twitter API.

As the creation of a large corpus of manually-labeled data for sentiment classification tasks involves significant human effort, a number of studies has explored the use of emoticons as labels [7, 5, 22]. The use of emoticons assumes that they could be associated with positive and negative polarities regarding the subject mentioned in the tweet. Although there are cases where this basic assumption holds, there are some cases where the relation between the emoticon and the tweet subject is not clear. Hence, the use of emoticons as tweet's labels can introduce noise. However, this drawback is counterweighted by the large amount of data that can easily be labeled. In this direction, Go et al. [7] reported the creation of a large Twitter dataset with more than $1,600,000$ tweets. By using standard machine learning algorithms, accuracies greater than 80% were reported for label prediction. Recently, Liu et al. [11] explored the combination of emoticon labels and human labeled tweets in language models, outperforming previous approaches.

Sentiment Lexical resources were used as features in a supervised classification scheme in [10, 9, 26] among other works. In [10] a supervised approach for Twitter sentiment classification based on linguistic features was proposed. In addition of using n-grams and part-of-speech tags as features, the authors used sentiment lexical resources and aspects particular from microblogging platforms such as the presence of emoticons, abbreviations and intensifiers. A comparison of the different types of features was carried out, showing that although features created from the opinion lexicon are relevant, microblogging-oriented features are the most useful.

Recently, The Semantic Evaluation (SemEval) workshop has organized a Sentiment Analysis in Twitter task (SemEval-2013)[2]. This task provides training and testing datasets for Twitter sentiment classification at both expression and message levels [24].

For further details about sentiment analysis methods and applications we refer the reader to the survey of Pang and Lee [20] and to the book of Liu [12].

[1] http://sentic.net/

[2] http://www.cs.york.ac.uk/semeval-2013/task2/

3. CLASSIFICATION APPROACH

In this section we describe the proposed Twitter sentiment classification approach. We consider two classification tasks: subjectivity and polarity classification. In the former, tweets are classified as subjective (non neutral) or objective (neutral), and in the latter as positive or negative. Moreover, positive and negative tweets are considered as subjective.

We propose a supervised approach for which we model each tweet as a vector of sentiment features. Additionally, a dataset of manually annotated tweets is required for training and evaluation purposes. Once the feature vectors of all the tweets from the dataset have been extracted, they are used together with the annotated sentiment labels as input for supervised learning algorithms. Several learning algorithms can be used to fullfill this task, eg. naive Bayes, SVM, decision trees. Finally, the resulting learned function can be used to infer automatically the sentiment label regarding an unseen tweet. All the resources and methods considered in this work are publicly available, facilitating repeatability of our experiments.

In contrast to the common text classification approach, in which the words contained within the passage are used as features (e.g., unigrams, n-grams), our meta-level features are based on existing lexical resources and sentiment analysis methods. These resources and methods, summarize the main efforts discussed in Section 2, and cover three different dimensions of the problem: polarity, strength, and emotions.

From each lexical resource we calculate a number of features according to the number of matches between the words from the tweet and the words from the lexicon. If the lexical resource provides strength values associated to the words, then features are calculated through a weighted sum. Finally, for each sentiment analysis method, its outcome is included as a dimension in the feature vector. The features are summarized in Table 1, and are described together with their respective methods and resources in the following parapraphs.

OpinionFinder Lexicon.

The **OpinionFinder Lexicon** (**OPF**) is a polarity oriented lexical resource created by Wilson et al. [25]. It is an extension of the Multi-Perspective Question-Answering dataset (MPQA), that includes phrases and subjective sentences. A group of human annotators tagged each sentence according to the polarity classes: positive, negative, neutral[3]. Then, a pruning phase was conducted over the dataset to eliminate tags with low agreement. Thus, a list of sentences and single words was consolidated, with their polarity tags. In this study we consider single words (unigrams) tagged as positive or negative, that correspond to a list of $6,884$ English words. We extract from each tweet two features related to the OpinionFinder lexicon, **OpinionFinder Positive Words** (OPW) and **OpinionFinder Negative Words** (ONW), that are the number of positive and negative words of the tweet that matches the OpinionFinder lexicon, respectively.

AFINN Lexicon.

This lexicon is based on the **Affective Norms for English Words** lexicon (ANEW) proposed by Bradley and Lang [3]. ANEW provides emotional ratings for a large number of English words. These ratings are calculated according to the psychological reaction of a person to a specific word, being "valence" the most useful value for sentiment analysis. "Valence" ranges in the scale pleasant-unpleasant. ANEW was released before the rise of microblogging and hence, many slang words commonly used in social media were not included. Considering that there is empirical evidence about significant differences between microblogging words and the language used in other domains [2] a new version of ANEW was required. Inspired in ANEW, Nielsen [16] created the **AFINN** lexicon, which is more focused on the language used in microblogging platforms. The word list includes slang and obscene words as also acronyms and web jargon. Positive words are scored from 1 to 5 and negative words from -1 to -5, reason why this lexicon is useful for strength estimation. The lexicon includes $2,477$ English words. We extract from each tweet two features related to the AFINN lexicon, **AFINN Positivity** (APO) and **AFINN Negativity** (ANE), that are the sum of the ratings of positive and negative words of the tweet that matches the AFINN lexicon, respectively.

SentiWordNet Lexicon.

SentiWordNet 3.0 (**SWN3**) is a lexical resource for sentiment classification introduced by Baccianella et al. [1], that it is an improvement of the original SentiWordNet proposed by Esuli and Sebastiani [6]. SentiWordNet is an extension of **WordNet**, the well-known English lexical database where words are clustered into groups of synonyms known as **synsets** [13]. In SentiWordNet each synset is automatically annotated in the range $[0, 1]$ according to positivity, negativity and neutrality. These scores are calculated using semi-supervised algorithms. The resource is available for download[4]. In order to extract strength scores from SentiWordNet, we use the word's scores to compute a real value from -1 (extremely negative) to 1 (extremely positive), where neutral words receive a zero score. We extract from each tweet two features related to the SentiWordnet lexicon, **SentiWordnet Positiveness** (SWP) and **SentiWordnet Negativeness** (SWN), that are the sum of the scores of positive and negative words of the tweet that matches the SentiWordnet lexicon, respectively.

SentiStrength Method.

SentiStrength is a lexicon-based sentiment evaluator that is specially focused on short social web texts written in English [23]. SentiStrength considers linguistic aspects of the passage such as a negating word list and an emoticon list with polarities. The implementation of the method can be freely used for academic purposes and is available for download[5]. For each passage to be evaluated, the method returns a positive score, from 1 (not positive) to 5 (extremely positive), a negative score from -1 (not negative) to -5 (extremely negative), and a neutral label taking the values: -1 (negative), 0 (neutral), and 1 (positive). We extract from each tweet three features related to the SentiStrength method, **SentiStrength Negativity** (SSN) and **SentiStrength Positivity** (SSP), that correspond to the strength scores for the negative and positive classes, respectively, and **SentiStrength Polarity** (SSPOL), that is a polarity-oriented feature corresponding to the neutral label.

Sentiment140 Method.

Sentiment140[6] is a Web application that classifies tweets according to their polarity. The evaluation is performed using the distant supervision approach proposed by Go et al. [7] that was previously discussed in the related work section. The approach relies on supervised learning algorithms and due to the difficulty of obtaining a large-scale training dataset for this purpose, the problem is tackled using positive and negative emoticons and noisy labels. The

[3]The lexicon also includes 17 words having mixed positive and negative tags tagged as "both", which were omitted in this work.

[4]http://sentiwordnet.isti.cnr.it/

[5]http://sentistrength.wlv.ac.uk/

[6]http://www.sentiment140.com/

Scope	Feature	Source	Description	Range
Polarity	SSPOL	SentiStrength	method label (negative, neutral, positive)	$\{-1, 0, +1\}$
	S140	Sentiment140	method label (negative, neutral, positive)	$\{-1, 0, +1\}$
	OPW	OpinionFinder	number of positive words that matches OpinionFinder	$\{0, 1, ..., n\}$
	ONW		number of negative words that matches OpinionFinder	$\{0, 1, ..., n\}$
Strength	SSP	SentiStrength	method score for the positive category	$\{1, \ldots, 5\}$
	SSN		method score for the negative category	$\{-5, \ldots, -1\}$
	SWP	SentiWordNet	sum of the scores for the positive words that matches the lexicon	$\{0, \ldots, n\}$
	SWN		sum of the scores for the negative words that matches the lexicon	$\{0, \ldots, n\}$
	APO	AFINN	sum of the scores for the positive words that matches the lexicon	$\{0, ..., n\}$
	ANE		sum of the scores for the negative words that matches the lexicon	$\{-n, ..., 0\}$
Emotion	NJO	NRC	number of words that matches the joy word list	$\{0, 1, ..., n\}$
	NTR		... matches the trust word list	$\{0, 1, ..., n\}$
	NSA		... matches the sadness word list	$\{0, 1, ..., n\}$
	NANG		... matches the anger word list	$\{0, 1, ..., n\}$
	NSU		... matches the surprise word list	$\{0, 1, ..., n\}$
	NFE		... matches the fear word list	$\{0, 1, ..., n\}$
	NANT		... matches the anticipation word list	$\{0, 1, ..., n\}$
	NDIS		... matches the disgust word list	$\{0, 1, ..., n\}$

Table 1: Features can be grouped into three classes having as scope Polarity, Strength, and Emotion, respectively.

method provides an API[7] that allows to classify tweets to polarity classes positive, negative and neutral. We extract from each tweet one feature related to the Sentiment140 output, **Sentiment140** class (S140), that corresponds to the output returned by the method.

NRC Lexicon.

NRC is a lexicon that includes a large set of human-provided words with their emotional tags. By conducting a tagging process in the crowdsourcing Amazon Mechanical Turk platform, Mohammad and Turney [14] created a word lexicon that contains more than 14,000 distinct English words annotated according to the Plutchik's wheel of emotions. These words can be tagged to multiple categories. Eight emotions were considered during the creation of the lexicon, joy-trust, sadness-anger, surprise-fear, and anticipation-disgust, which compounds four opposing pairs. Additionally, NRC words are tagged according to polarity classes positive and negative, which are not considered in this work. The word list is available under request[8]. We extract from each tweet eight features related to the NRC lexicon, **NRC Joy** (NJO), **NRC Trust** (NTR), **NRC Sadness** (NSA), **NRC Anger** (NANG), **NRC Surprise** (NSU), **NRC Fear** (NFE), **NRC Anticipation** (NANT), and **NRC Disgust** (NDIS), that are the number of words of the tweet that matches each category.

4. EXPERIMENTS

4.1 Lexical Resource Interaction

In this section we study the interaction of words between the different lexical resources: SWN3, NRC, OpinionFinder, and AFINN. The number of words that overlap between each pair of resources is shown in Table 2. From the table we can see that SWN3 is much larger than the other resources. Nevertheless, the resource includes many neutral words provided by WordNet that lack of useful information for sentiment analysis purposes.

Table 3 shows the overlap of words after discarding the neutral words from SentiWordNet, the neutral and mixed words from OpinionFinder and the words without emotion tags from NRC. We can see that although the size of SWN3 was strongly reduced it stills has much more words than the others. The interaction of all the non-neutral words, is better represented in the Venn diagram shown in Figure 1. From the diagram we can see that SWN3 covers the majority of the words within the lexical resources. However, if we discard SWN3 we keep with three different sets of words: NRC having words related to emotions, OpinionFinder whose words are related to polarity, and AFINN whose words are also related to polarity with additional strength information. These resources, in addition to having different sentiment scopes, cover many different words from each other. It is also revealed from the figure that the AFINN lexicon, despite being smaller, contains some words that are not included in SWN3 nor in the others. We inspected these words included only in AFINN and we found many Internet acronyms and slang words such as "lmao", "lol", "rofl", "wtf" among other expressions.

We compare the sentiment values assigned by each lexical resource to a sample of words that appear in the intersection of all lexicons in Table 4. We can observe a tendency of the different resources to support each other, eg. words that received negative strength values from SWN3 and AFINN normally receive a nega-

	SWN3	NRC	AFINN	OPFIND
SWN3	147,306	×	×	×
NRC	13,634	14,182	×	×
AFINN	1,783	1,207	2,476	×
OPFIND	6,199	3,596	1,245	6,884
Distinct Words	149,114			

Table 2: Intersection of words between different Lexical Resources

	SWN3	NRC	AFINN	OPFIND
SWN3	33,313	×	×	×
NRC	2,932	3,071	×	×
AFINN	1,203	721	1,871	×
OPFIND	3,703	1,658	900	4,311
Distinct Words	34,649			

Table 3: Intersection of non-neutral words

[7] http://help.sentiment140.com/api

[8] mailto: saif.mohammad@nrc-cnrc.gc.ca

word	SWN3	AFINN	OPFIND	NRC
abuse	-0.51	-3	negative	ang,disg,fear,sadn
adore	0.38	3	positive	ant, joy, trust
cheer	0.13	2	positive	ant, joy, surp, trust
shame	-0.52	-2	negative	digs,fear,sadn
stunned	-0.31	-2	positive	fear, surpr
sympathy	-0.13	2	negative	sadn
trust	0.23	1	positive	trust
ugly	-0.63	-3	negative	disg
wonderful	0.75	4	positive	joy, surp, trust

Table 4: Sentiment Values of Words included in all the Resources

tive tag from OpinionFinder and are associated as well with negative NRC emotions states. A similar pattern is observed for positive words. However, we can also see controversial examples such as words "stunned" and "sympathy" which receive contrary sentiment values from polarity and strength oriented resources. These words may be used to express either positive and negative opinions, depending on the context. Considering that it is very hard to associate them to a single polarity class, we think that emotion tags explain in a better manner the diversity of sentiment states triggered by these kind of words.

These insights indicate that the resources considered in this work complement each other, providing different sentiment information.

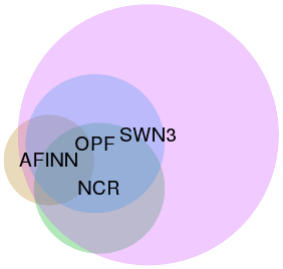


Figure 1: Non-neutral words interaction Venn diagram

4.2 Training and Testing Datasets

We consider two collections of tweets for our experiments: *Stanford Twitter Sentiment* (STS) [9] which was used by Go et al. [7] in their experiments, and *Sanders*[10]. Each tweet includes a **positive**, **negative** or **neutral** tag. Table 5 summarizes both datasets.

Negative and positive tweets were considered as subjective. Neutral tweets were considered as objective. Subjective/objective tags favor the evaluation of subjectivity detection. For polarity detection tasks, positive and negative tweets were considered, discarding neutral tweets.

Both datasets were balanced. Class imbalance was tackled by sampling 139 subjective tweets in STS from the 359 positive and negative tagged tweets, achieving a balance with the 139 neutral tweets. In the case of Sanders, the neutral collection was sampled recovering 1,196 tweets from the 2,429 neutral tweets achieving a balance with the 1,196 positive and negative tagged tweets. A similar process was conducted for class imbalance in the case of polarity recovering 354 and 1,120 tweets from STS and Sanders respectively. Table 6 summarizes the balanced datasets.

[9] http://cs.stanford.edu/people/alecmgo/trainingandtestdata.zip

[10] http://www.sananalytics.com/lab/twitter-sentiment/

	STS	Sanders
#negative	177	636
#neutral	139	2,429
#positive	182	560
#total	498	3,625

Table 5: Datasets Statistics

Subjectivity	STS	Sanders
#neutral	139	1,196
#subjective	139	1,196
#total	278	2,392
Polarity	**STS**	**Sanders**
#negative	177	560
#positive	177	560
#total	354	1,120

Table 6: Balanced Datasets

4.3 Feature Analysis

For each tweet of the two datasets we calculated the features summarized in Table 1. In a first analysis we explored how well each feature splits each dataset regarding polarity and subjectivity detection tasks. We do this by calculating the information gain criterion of each feature in each category. The information gain criterion measures the reduction of the entropy within each class after performing the best split induced by the feature. Table 7 shows the information gain values obtained.

Scope	Feature	Subjectivity		Polarity	
		STS	Sanders	STS	Sanders
Polarity	SSPOL	**0.179**	0.089	**0.283**	**0.192**
	S140	**0.103**	0.063	**0.283**	**0.198**
	OPW	0.088	0.024	0.079	0.026
	ONW	0.097	0.024	**0.135**	0.075
Strength	SSP	0.071	0.037	**0.200**	**0.125**
	SSN	0.090	0.044	**0.204**	**0.118**
	SWN	0.090	0.023	**0.147**	0.089
	SWP	**0.104**	0.030	0.083	0.015
	APO	0.088	0.024	0.079	0.026
	ANE	**0.134**	0.048	**0.200**	**0.143**
Emotion	NJO	0.000	0.000	0.055	0.065
	NTR	0.000	0.000	0.000	0.000
	NSA	0.000	0.017	0.000	0.056
	NANG	0.000	0.016	0.046	0.055
	NSU	0.000	0.000	0.000	0.017
	NFE	0.000	0.008	0.039	0.024
	NANT	0.000	0.000	0.000	0.000
	NDIS	0.000	0.014	0.056	0.030

Table 7: Feature information gain for each sentiment analysis task. Bold fonts indicate the best splits.

As Table 7 shows, the best polarity splits are achieved by using the outcomes of the methods(see SSPOL, S140, SSP, and SSN). SentiWordNet, OpinionFinder and AFINN-based features are useful for negative polarity detection. These features are also useful for subjectivity detection. In addition, we can observe that the best splits are achieved in the STS. The Sanders dataset is hard to split.

By analyzing the scope, we can observe that polarity-based features are the most informative. This fact is intuitive because the target variables belong to the same scope. Finally, although emotion features provide almost no information for subjectivity, some of them like joy, sadness and disgust are able to provide some information for the polarity classification task.

We also explored feature-subsets extracted by the correlation feature selection algorithm (CFS) [8]. This algorithm is a best-first feature selection method that considers different types of correlation as selection criteria. Selected features for each classification task on the two datasets are displayed in Table 8.

	Neu.STS	Neu.San	Pol.STS	Pol.San
ANE	✓	✓	✓	✓
APO	✓		✓	✓
ONW	✓		✓	✓
OPW	✓			
NJO				✓
S140	✓	✓	✓	✓
SSN			✓	✓
SSP			✓	
SSPOL	✓	✓	✓	✓
SWN	✓		✓	✓
SWP	✓	✓		

Table 8: Selected Features by CFS algorithm

From the table we can see that the two features that come from polarity-oriented methods (S140 and SSPOL), are selected in all the cases. We can also observe that the algorithm tends to include more features for polarity than for subjectivity classification in the Sanders dataset. Regarding the emotion-oriented features, the only feature that is selected by the CFS algorithm is the NJO feature. Moreover, the feature is only selected for the polarity task on the Sanders dataset. These results agree with the information gain values discussed above, and support the evidence that most of the features are more informative for polarity than for subjectivity classification.

4.4 Classification Results

We evaluate a number of learning algorithms on the STS and Sanders datasets, for both subjectivity and polarity detection. We conducted a 10-fold cross-validation evaluation. As learning algorithms we considered CART, J48, Naive Bayes, Logistic regression, and RBF SVMs. The experiments were performed using R 2.15.2 packages using the following packages: **rpart**[11] for CART, **rWeka**[12] for J48 and Logistic regression, and **e1071**[13] for Naive Bayes and SVMs.

The performance of many machine learning techniques are highly dependent on the calibration of parameters. Different parameters such as the min-split criterion for trees, γ and C for radial SVMs, among others were tuned using a grid-search procedure with nested 10-fold cross validation.

An example of the tuning process for the radial SVM for polarity classification on the Sanders dataset is shown in Figure 2. The x-axis and y-axis of the chart represent the **gamma** and **cost** parameter respectively. The color of the region corresponds to the classification error obtained using the corresponding parameter values. From the figure we can see that the classification performance varies considerably for different parameters values. Therefore, it is important to remark that the tuning process of machine learning parameters is crucial to obtain accurate classifiers.

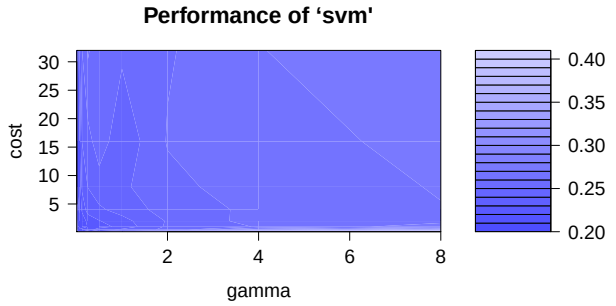


Figure 2: RBF SVM parameters performance for Polarity classification on Sanders dataset

A relevant issue regarding our feature-set is its heterogeneity. Most of the features are numerical but two of them are categorical (S140 and SSPOL). A number of supervised learning algorithms are not capable to handle mixed-type features and hence some transformations must be applied before the learning task. For CART and J48 the numerical features are "discretized" as part of the learning process. Naive Bayes handles numerical features by assuming Gaussian distributions. For the SVM and Logistic regression algorithms, we transformed categorical features into dummy variables by mapping the c possible categories to binary values using 1-of-c encoding. Afterwards, these binary variables are handled as numerical features by these learning algorithms.

The performance of our classifiers in both classification tasks is compared with baselines created from isolated methods or resources. In the subjectivity task we considered the features Sent140 and SSPOL as the **Baseline.1** and **Baseline.2**, respectively. For both methods the positive and negative outputs are interpreted as subjective. We chose these features because they are the only ones which explicitly distinguish between subjective and neutral tweets.

Nevertheless, these methods could not be used as baselines for the polarity task, because it is not clear how to handle their neutral outcomes in this context. Therefore, we created two categorical variables whose outcomes are restricted to **positive** and **negative** values. The **Baseline.1** is calculated from strength features SSP and SSN as follows: if the sum of **SSP** and **SSN** is positive the baseline takes a **positive** value, otherwise takes a **negative** value. Then, the second baseline (**Baseline.2**), is calculated in the same manner from the features **APO** and **ANE**. Considering that for SentiStrength and AFINN, positivity and negativity are assessed independently, basically what we are doing in our baselines is to combine these dimensions into categorical variables that are constrained to distinguish between positive and negative tweets.

In addition of the feature-subset obtained by the best first CFS algorithm, we also explored feature-subsets constrained to the scope. Thus, we evaluate five groups of features -all, best first, polarity, strength, and emotion- and for each group five learning algorithms -CART, J48, naive Bayes, logistic regression, and SVMs-.

We consider as performance measures **accuracy**, **precision**, **recall** and F_1. We believe that the costs of misclassifying each type of observation for each classification task are equally important. Thus, considering that our datasets are balanced, we will pay more attention to the measures accuracy and F_1 than to precision and

[11] http://cran.r-project.org/web/packages/rpart/
[12] http://cran.r-project.org/web/packages/RWeka
[13] http://cran.r-project.org/web/packages/e1071/

Dataset		STS				Sanders			
Features	Methods	accuracy	precision	recall	F_1	accuracy	precision	recall	F_1
Baseline.1	Sent140	0.655	0.812	0.403	0.538	0.615	0.686	0.424	0.524
Baseline.2	SSPOL	0.734	0.712	0.784	0.747	0.659	0.632	0.760	0.690
All	CART	0.694	0.696	0.691	0.693	0.686	0.688	0.683	0.685
	J48	0.716	0.742	0.662	0.700	0.694	0.703	0.673	0.688
	Naive Bayes	0.737	0.784	0.655	0.714	0.649	0.718	0.491	0.583
	Logistic	0.755	0.775	0.719	0.746	0.678	0.679	0.675	0.677
	SVM	0.763	0.766	0.755	0.761	**0.701**	0.696	0.713	0.705
Best.First	CART	0.730	0.735	0.719	0.727	0.677	0.639	**0.816**	**0.717**
	J48	0.701	0.730	0.640	0.682	0.673	0.639	0.796	0.709
	Naive Bayes	0.759	0.821	0.662	0.733	0.651	**0.727**	0.483	0.581
	Logistic	0.748	0.756	0.734	0.745	0.683	0.676	0.704	0.690
	SVM	0.773	0.757	**0.806**	**0.780**	0.680	0.663	0.732	0.696
Polarity	CART	0.734	0.712	0.784	0.747	0.677	0.639	**0.816**	**0.717**
	J48	0.676	0.684	0.655	0.669	0.673	0.639	0.797	0.709
	Naive Bayes	0.748	0.772	0.705	0.737	0.671	0.688	0.625	0.655
	Logistic	0.748	0.767	0.712	0.739	0.676	0.656	0.742	0.696
	SVM	0.759	0.765	0.748	0.756	0.674	0.637	0.810	0.713
Strength	CART	0.719	0.729	0.698	0.713	0.661	0.653	0.686	0.669
	J48	0.701	0.697	0.712	0.705	0.646	0.628	0.716	0.669
	Naive Bayes	0.766	**0.830**	0.669	0.741	0.636	0.711	0.460	0.558
	Logistic	0.763	0.797	0.705	0.748	0.662	0.688	0.593	0.637
	SVM	**0.777**	0.824	0.705	0.760	0.694	0.683	0.725	0.703
Emotion	CART	0.579	0.634	0.374	0.471	0.586	0.638	0.398	0.490
	J48	0.590	0.647	0.396	0.491	0.575	0.628	0.370	0.465
	Naive Bayes	0.579	0.628	0.388	0.480	0.573	0.647	0.320	0.428
	Logistic	0.583	0.624	0.417	0.500	0.585	0.635	0.402	0.492
	SVM	0.597	0.622	0.496	0.552	0.594	0.627	0.462	0.532

Table 9: 10-fold Cross-Validation Subjectivity Classification Performances

recall measures, as was also done in [11]. This is because accuracy and F_1 measures are affected by both false positive and false negative results.

Table 9 shows the results for the subjectivity classification task. We can observe that **Baseline.2** outperforms **Baseline.1** in both datasets. This is because Sentiment140 is not focused on subjectivity classification.

There are significant performance differences between both datasets. We hypothesize that STS's tweets have good properties for classification because they show clear differences between neutral and non neutral tweets. On the other hand, in the Sanders dataset, we found tweets marked as neutral that contain mixed positive and negative opinions. Two examples of this kind of tweets are presented below.

1. *Hey @Apple, pretty much all your products are amazing. You blow minds every time you launch a new gizmo. That said, your hold music is crap.*
2. *#windows sucks... I want #imac so bad!!! why is it so damn expensive :(@apple please give me free imac and I will love you :D*

Both tweets are about the company **Apple**. The first tweet shows a positive opinion about Apple's products and at the same time shows a negative opinion about Apple's hold music. This example contains contrary opinions about two different aspects of the entity Apple. The second example is even more complicated because it expresses opinions on two different entities: **Windows** and **Apple**. The tweet compares two products and shows a clear preference for Apple's product **iMac**. Additionally, the message indicates that the product **iMac** is too expensive, something that could be interpreted as a negative opinion about the product. By inspection, that kind of tweets are not included in STS. Due to this fact, we believe that in addition of being larger, Sanders captures in a better way than the STS corpus the sentiment diversity of tweets. Nevertheless, considering that tweets with mixed positive and negative indicators are subjective, we believe that labeling them as **neutral** may increase the level of noise in the data.

Regarding learning algorithms, SVM tends to outperform other methods in accuracy and F_1, and most of the best results are achieved using the best feature selection algorithm. As was expected, the emotion feature subset achieves poor classification results for this task.

Polarity performance results are showed in Table 10. In this case, both baselines are strongly competitive, being the SentiStrength-based baseline better than the other one. This result agrees with the results reported by Nielsen [16] where it was shown that the AFINN lexicon was not able to outperform SentiStrength. We can observe also that the detection of polarity is a more difficult task in Sanders than in STS, as was also observed for the subjectivity detection task.

The best tree obtained for polarity classification by the CART algorithm using all the features on the Sanders dataset is shown in Figure 3. From the figure with can see that top level nodes of the tree correspond to features related to SentiStrength, Sentiment140 and AFINN. This results correspond with the information gain values obtained and explains in some manner why these methods are competitive as baselines. The tree also indicates that negative words from the different lexical resources are more useful than the positive ones.

In a similar way as in the subjectivity task, SVM achieves the best results in accuracy and F_1. This fact suggests that there are non-linearities between the features that are successfully tackled by using the RBF kernel. The performance tends also in both datasets to be better for the polarity task than for the subjectivity problem. This is because most of the lexical resources and methods are more

Dataset		STS				Sanders			
Features	Methods	accuracy	precision	recall	F_1	accuracy	precision	recall	F_1
Baseline.1	SentiStrength	0.777	0.766	0.797	0.781	0.733	0.735	0.729	0.732
Baseline.2	AFINN	0.771	0.804	0.718	0.758	0.713	0.747	0.643	0.691
All	CART	0.788	0.790	0.785	0.788	0.780	0.759	0.821	0.789
	J48	0.788	0.768	0.825	0.796	0.775	0.769	0.786	0.777
	Naive Bayes	0.794	0.757	0.864	0.807	0.774	0.729	0.873	0.794
	Logistic	0.805	0.784	0.842	0.812	**0.801**	0.782	0.834	0.807
	SVM	0.808	**0.808**	0.808	0.808	**0.801**	0.775	0.848	**0.810**
Best.First	CART	0.791	0.775	0.819	0.797	0.789	**0.790**	0.788	0.789
	J48	0.802	0.789	0.825	0.807	0.781	0.778	0.788	0.783
	Naive Bayes	0.811	0.775	**0.876**	0.822	0.788	0.750	0.863	0.802
	Logistic	0.814	0.803	0.831	0.817	0.778	0.765	0.802	0.783
	SVM	**0.816**	0.795	0.853	**0.823**	0.792	0.760	0.854	0.804
Polarity	CART	0.802	0.796	0.814	0.804	0.779	0.736	0.870	0.797
	J48	0.791	0.764	0.842	0.801	0.775	0.728	0.877	0.796
	Naive Bayes	0.805	0.787	0.836	0.811	0.756	0.736	0.800	0.766
	Logistic	0.799	0.779	0.836	0.807	0.786	0.771	0.813	0.791
	SVM	0.799	0.770	0.853	0.810	0.776	0.728	0.882	0.797
Strength	CART	0.780	0.783	0.774	0.778	0.705	0.686	0.757	0.720
	J48	0.777	0.772	0.785	0.779	0.746	0.732	0.775	0.753
	Naive Bayes	0.780	0.746	0.847	0.794	0.762	0.711	0.880	0.787
	Logistic	0.797	0.800	0.791	0.795	0.752	0.747	0.761	0.754
	SVM	0.799	0.805	0.791	0.798	0.779	0.747	0.845	0.793
Emotion	CART	0.684	0.637	0.853	0.729	0.658	0.630	0.766	0.691
	J48	0.681	0.629	0.881	0.734	0.650	0.620	0.777	0.689
	Naive Bayes	0.641	0.599	0.853	0.704	0.654	0.604	**0.891**	0.720
	Logistic	0.661	0.623	0.814	0.706	0.671	0.637	0.795	0.707
	SVM	0.624	0.598	0.757	0.668	0.656	0.624	0.784	0.695

Table 10: 10-fold Cross-Validation Polarity Classification Performances

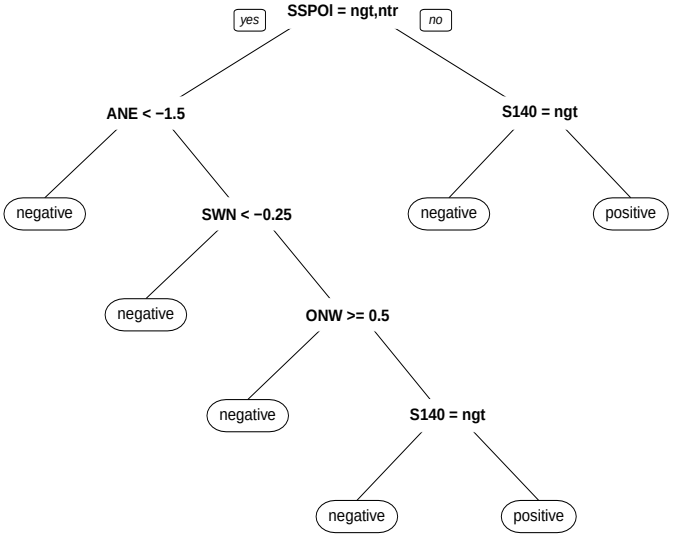


Figure 3: Best Tree trained with CART for polarity classification on the Sanders dataset

focused on the detection of polarity rather than detecting subjectivity.

As was discussed before, emotion-oriented features tend to have low information gain values and also present a poor classification performance. Therefore, it would make sense to think that emotion-oriented features are not useful for sentiment classification. However, if we consider the accuracies obtained by RBF SVMs on the Sanders dataset for both classification tasks, we can see that in addition of outperforming the others learning algorithms, they achieved the best accuracies when all type of features were included. That means, that emotion-oriented features are useful for sentiment classification when they are combined with polarity and strength oriented features in a non-linear fashion.

The best learned functions obtained for each classification task outperformed the results achieved by the baselines created from isolated methods. Thus, our results validate the hypothesis that the combinations of different sentiment analysis methods and resources enhances the overall sentiment classification.

5. CONCLUSIONS AND FUTURE WORK

We present a novel approach for sentiment classification on microblogging messages or short texts based on the combination of several existing lexical resources and sentiment analysis methods. Our experimental validation shows that our classifiers achieve very significant improvements over any singe method, outperforming state-of-the-art methods by more than 5% accuracy and F_1 points.

Considering that the proposed feature representation does not depend directly on the vocabulary size of the collection, it provides a considerable dimensionality reduction in comparison to word-based representations such as unigrams or n-grams. Likewise, our approach also avoids the sparsity problem presented by word-based feature representations for Twitter sentiment classification discussed in [15]. Due to this, our low-dimensional feature representation allows us to efficiently use several learning algorithms.

The classification results varied significantly from one dataset to another. The manual sentiment classification of tweets is a subjective task that can be biased by the evaluator's perceptions. This fact should serve as a warning call against bold conclusions from inadequate evidence in sentiment classification. It is very important to check beforehand whether the labels in the training dataset correspond to the desired values, and if the training examples are able to capture the sentiment diversity of the target domain.

Finally, it is important to recall that opinions are multidimensional objects. In this way, when we classify tweets into polarity classes, we are essentially projecting these multiple dimensions to one single categorical dimension. Furthermore, it is not clear how to project tweets having mixed positive and negative expressions to a single polarity class. Therefore, we have to be aware that the sentiment classification of tweets may lead to the loss of valuable sentiment information.

As future work we expect to expand this study by including other sentiment resources and methods which were not considered at this moment. For instance we expect to create semantic-level features from concept-based resources such as *SenticNet*. Additionally, we plan to evaluate our approach on the *SemEval* task datasets in order to compare our results with other works that participated in the task.

6. ACKNOWLEDGMENT

This work has been partially supported by FONDEF-CONICYT project D09I1185. Felipe Bravo-Marquez was supported by CONICYT's Master Scholarship. Marcelo Mendoza was supported by project FONDECYT 11121435. Barbara Poblete was partially supported by FONDECYT grant 11121511 and Program U-INICIA VID 2012, grant U-INICIA 3/0612; University of Chile.

7. REFERENCES

[1] Baccianella, S., Esuli, A., and Sebastiani, F. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, 2010. Valletta, Malta.

[2] Baeza-Yates, R., and Rello, L. How Bad Do You Spell?: The Lexical Quality of Social Media. The Future of the Social Web, Papers from the 2011 ICWSM Workshop, Barcelona, Catalonia, Spain. AAAI Workshops, 2011.

[3] Bradley, M. M., and Lang, P. J. Affective Norms for English Words (ANEW) Instruction Manual and Affective Ratings. *Technical Report C-1, The Center for Research in Psychophysiology* University of Florida, 2009.

[4] Cambria, E., Speer R., Havasi C., and Hussain A. SenticNet 2: A Semantic and Affective Resource for Opinion Mining and Sentiment Analysis. In *FLAIRS Conference*, pages 202–207, 2012.

[5] Carvalho, P., Sarmento, L., Silva, M. J., and de Oliveira, E. Clues for detecting irony in user-generated contents: oh...!! it's "so easy" ;-). In *Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion* Hong Kong, China, 2009.

[6] Esuli, A., and Sebastiani, F. Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation, 2006*.

[7] Go, A., Bhayani, R., and Huang, L. Twitter sentiment classification using distant supervision. *Technical report Stanford University*, 2010.

[8] Hall, M. *Correlation-based Feature Selection for Machine Learning*. PhD thesis, University of Waikato, 1999.

[9] Jiang, L., Yu, M., Zhou, M., Liu, X., and Zhao, T. Target-dependent twitter sentiment classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 151–160, 2011.

[10] Kouloumpis, E., Wilson, T., and Moore, J. Twitter Sentiment Analysis: The Good the Bad and the OMG! In *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[11] Liu, K., Li, W., and Guo, M. Emoticon smoothed language models for Twitter sentiment analysis. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*. Toronto, Ontario, Canada, 2012.

[12] Liu, B. Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies series, Morgan & Claypool Publishers, 2012.

[13] Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244, 1990.

[14] Mohammad, S. M., and Turney, P. D. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 2012.

[15] Saif, H., He, Y., and Alani, H. Alleviating data sparsity for twitter sentiment analysis. In *Workshop of Making Sense of Microposts co-located with WWW 2012*, 2012.

[16] Nielsen, F. Å. A new anew: Evaluation of a word list for sentiment analysis in microblogs. In *ESWC2011 Workshop on Making Sense of Microposts*, May 2011.

[17] Grassi, M., Cambria, E., Hussain, A., and Piazza, F. Sentic web: A new paradigm for managing social media affective information. *Cognitive Computation*, 3(3):480–489, 2011.

[18] Olsher, D J. Full spectrum opinion mining: Integrating domain, syntactic and lexical knowledge. In *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*, pages 693–700. IEEE, 2012.

[19] Pak, A., and Paroubek, P. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta, 2010.

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

[21] Plutchik, R. 2002. Nature of emotions, American Scientist, 89, 349.

[22] Read, J. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. Michigan, USA, 2005.

[23] Thelwall, M., Buckley, K., and Paltoglou, G. Sentiment strength detection for the social web. *JASIST* 63(1):163–173. 2012.

[24] Wilson, T., Kozareva, Z., Nakov, P., Ritter A., Rosenthal, S., and Stoyonov V. Semeval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the 7th International Workshop on Semantic Evaluation*. Association for Computation Linguistics, 2013.

[25] Wilson, T., Wiebe, J., and Hoffmann, P. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*. Vancouver, British Columbia, Canada, 2005.

[26] Zirn, C., Niepert M., Stuckenschmidt H., and Strube M. Fine-grained sentiment analysis with structural features. In *IJCNLP*, pages 336–344, 2011.

Modelling Political Disaffection from Twitter Data

Corrado Monti
Dipartimento di Informatica, Università degli Studi di Milano
me@corradomonti.com

Alessandro Rozza
Dipartimento di Scienze Applicate, Università degli Studi di Napoli "Parthenope"
alessandro.rozza@uniparthenope.it

Giovanni Zappella
Dipartimento di Matematica, Università degli Studi di Milano
giovanni.zappella@unimi.it

Matteo Zignani
Dipartimento di Informatica, Università degli Studi di Milano
matteo.zignani@unimi.it

Adam Arvidsson
Dipartimento di Scienze Politiche, Università degli Studi di Milano
adam.arvidsson@unimi.it

Elanor Colleoni
Department of Intercultural Communication and Management, Copenhagen Business School
elc.ikl@cbs.dk

ABSTRACT

Twitter is one of the most popular micro-blogging services in the world, often studied in the context of political opinion mining for its peculiar nature of online public discussion platform. In our work we analyse the phenomenon of *political disaffection* defined as the "lack of confidence in the political process, politicians, and democratic institutions, but with no questioning of the political regime". Disaffection for organised political parties and institutions has been object of studies and media attention in several Western countries. Especially the Italian case has shown a wide diffusion of this attitude. For this reason, we collect a massive database of Italian Twitter data (about 35 millions of tweets from April 2012 to October 2012) and we exploit scalable state-of-the-art machine learning techniques to generate time-series concerning the political disaffection discourse.

In order to validate the quality of the time-series generated, we compare them with indicators of political disaffection from public opinion surveys. We find political disaffection on Twitter to be highly correlated with the indicators of political disaffection in the public opinion surveys. Moreover, we show the peaks in the time-series are often generated by external political events reported on the main newspapers.

General Terms

Political disaffection; Classification; Twitter; Sentiment Analysis

1. INTRODUCTION

Twitter is one of the most popular micro-blogging services in the world. Micro-blogging allows the publication of short text messages, used to share all kinds of information; on Twitter, these messages are called "tweets" (their maximum length is 140 characters), and many millions of them are posted every day.

WISDOM'13 August 12, 2013, Chicago, USA

Twitter has proven to be a relevant data source to explore public sentiment trends ([4, 32]). Its content is easily available, and its flexible nature allows harvesting open conversations, public opinions, and news commentaries. Another crucial characteristic of Twitter is its timeliness; this peculiarity guarantees that tweets are related to a much narrower temporal window with respect to other user-generated texts, such as blogs.

Modelling trends from Twitter data has become a popular research task. Among such studies, those drawing attention to political topics are some of the most attractive, and in the last years a great deal of research works has focused on them.

In this study we concentrate on *political disaffection*, an important concept in political science. Political disaffection has been defined by Di Palma as "the subjective feeling of powerlessness, cynicism, and lack of confidence in the political process, politicians, and democratic institutions, but with no questioning of the political regime" [11]. In political science, levels of political disaffection are understood to relate to levels of political participation and, consequently they have important implications for the legitimacy of democratic political systems. This phenomenon has been significant since the 1960's in many Western countries and it has gained even more mass media attention after the 2008 Financial Crisis. These facts make the study of political disaffection pressing issue of contemporary studies of political behaviour. Some of the most relevant studies (i.e. [1, 25, 30]) are focused on Italy, since this political phenomenon has been particularly important in this country in the last half century. Moreover, the annual European Commission surveys on public opinion[1] confirm the relevance of this attitude among Italian citizens.

To our knowledge political disaffection has never been studied using Twitter data. In this work we propose an automatic approach to measure political disaffection using a massive collection of Twitter data from the Italian community. Our aim is the study of the relations between our measurement of political disaffection and political disaffection as measured by public opinion surveys.

In accordance with Di Palma we define political disaffection as negative sentiment towards the political system in general, rather than towards a particular politician, policy or issue. We operationalize this concept by defining expression of political disaffection tweets that have the following characteristics: 1) *political*, 2) *negative sentiment*, and 3) *generality*; where the last two features

[1] *http://ec.europa.eu/public_opinion*

capture the lack of confidence in the whole political system. Consequently the measurement of political disaffection in Twitter can be performed by a sequence of three tasks. First, we use a supervised methodology to extract a subset of political tweets from the universe of tweets. Second, we perform a sentiment analysis on political tweets to extract those with negative sentiment. Third, we automatically select the tweets that refer to politics or politicians in general, rather than specific political events or personalities.

By applying our approach to the Italian Twitter community, we can monitor the trend of the political disaffection. This allows us to relate and compare the Twitter disaffection time series to indicators of political disaffection in public opinion surveys and thereby to validate our operational measurement. Finally, we show that external events such as important political news from Italian newspapers are often correlated with peaks in the produced time-series. This paper is organized as follows: in Section 2 the related works are summarized; in Section 3 we describe the procedures used to collect the datasets employed to train our supervised methods, the approach to extract the overall set of tweets employed in our analysis, and we summarize the public opinion surveys used to validate the quality of our approach; in Section 4 we describe the overall methodology to extract the political disaffection tweets; in Section 5 we present the achieved results on the extracted political disaffection trends; finally Section 6 contains the conclusions of our work.

2. RELATED WORKS

In literature a great deal of research has focused on the analysis of different phenomena using the data of micro-blogging services. Among them, in [24] the authors explore the correlation between types of user engagement and events about celebrities using Twitter data. Furthermore, in [3] the authors propose an approach to predict the stock market trend by monitoring micro-blogging.

The most closely related works are those concerning concept-level sentiment analysis [7], short text conceptualization [8] and the investigation of political topics using Twitter data. In [4], the authors propose a method to extract different time series corresponding to the evolution of 6 emotional attributes (tension, depression, anger, vigour, fatigue, and confusion) called Profile of Mood States (POMS). The authors apply POMS to suggest that socio-economic agitations caused significant fluctuations of the mood levels.

One of the earliest papers discussing the feasibility of using Twitter data as a substitute of traditional public opinion surveys is [22]. The authors employ Opinion-Finder[2] to determine both a positive and a negative score for each tweet in their dataset. Then, raw numbers of positive and negative tweets regarding a given topic are used to compute a sentiment score. Subsequently, sentiment time series are created for different topics such as: presidential approval, consumer confidence, and US 2008 Presidential elections. According to the authors both consumer confidence and presidential approval public opinion surveys show correlation with the Twitter sentiment data computed with their approach. However, no correlation has been found between electoral public opinion surveys and Twitter sentiment data.

In [31] an analysis of the tweets related to different parties running for the German 2009 Federal election has been carried out. The authors show that the volume of tweets mentioning a party or a candidate accurately reflected the election results, suggesting a possible approach to perform an electoral prediction. Furthermore, in [18] a novel method that aims at predicting elections has been proposed. This approach relies both on Twitter data and on additional information such as the party a candidate belongs to, or incumbency. Bermingham *et al.* [2] improve the previous approaches by incorporating sentiment analysis to the prediction of the political election. The authors tested their method in the 2011 Irish General Election finding that the results are not competitive when compared with traditional public opinion surveys. Similar approaches are proposed in [29, 28]. Nevertheless, the possibility to perform an electoral prediction using Twitter data [21, 13] is still an open issue. For instance, in [21] the authors analyse the results of different elections and they conclude that Twitter data is only slightly better than chance when predicting elections.

For this reason we avoid to predict election outcome in terms of percentages of party support, but we evaluate the well-known political attitude of political disaffection by analysing Twitter data through machine learning techniques. In order to validate the quality of the information extracted from the Twitter data, we highlight the relations of this data with political disaffection as measured in public opinion surveys.

3. DATA EMPLOYED

To model political disaffection on Twitter, we have to consider how a relevant tweet should be defined. According to the aforementioned Di Palma's definition, in the following we identified the characteristics of the tweets that express political disaffection:

- **Political**: obviously the subject of the tweets regards what Di Palma indicates as "political process, politicians and institution".
- **Negative**: we capture the powerlessness and the lack of confidence in the political system by analysing the negative feeling towards the subject of the tweets.
- **General**: the last sentence in Di Palma's definition denotes that the tweet subject is not a particular element of the political system. As a consequence tweets regarding most of parties or the whole political class are intended to be general, while tweets addressing a specific politician or institution do not belong to this category.

Once defined the subject of our work, we developed a filtering mechanism based on machine learning techniques with the purpose of extracting tweets from which a political disaffection feeling emerges. This filtering mechanism was constituted by a sequence of three tasks directly linked with the above characteristics.

The first task concerns the detection and the extraction of political tweets; in the second we retrieved negative tweets from the political ones, while in the last task we detected general tweets from those resulting after the second task (for details see Section 4).

In order to apply supervised machine learning algorithm to the first and the second task, we needed a reliable and big enough dataset to train our classifiers.

3.1 Training Twitter Data (TwitterTrainData)

We built the training set by a 2-step procedure involving a semi-automatic search method that employs the Twitter API v1 and a labeling phase guided by experts. The collection phase began at the beginning of April 2012 and ended at the beginning of June 2012. We collected about $120,000$ of tweets and retweets. The selected tweets result from a geo-localized trending topic[3] search and a tar-

[2]Opinion-Finder is a system that performs subjectivity analysis, automatically identifying when opinions, sentiments, speculations and other private states are present in text.

[3]Trending topics are the most popular and talked-about words and phrases on Twitter for a specific time period.

	"positive"$= +1$	"negative"$= -1$
"political"$= +1$	$7'965$	$4'544$
"political"$= -1$	$15'831$	

Table 1: Label distribution of the TwitterTrainData after the majority voting.

geted search on political themes. In particular, at the end of each day we requested the top 10 trending topics of the Italian community. As most trending topics regard non-political arguments (i.e. celebrities, sports or viral hashtags), we selected the political content and a subset of the non-political. Furthermore, in order to have a more meaningful number of political tweets, we searched for tweets related to politicians, political news from Italian online newspapers and talk-shows. As query keywords we chose the Italian politicians' surname, parties and organizations, the topics of the top news in the political section of online newspapers and the official hashtags of TV-talks. The resulting dataset consists of a large corpus of about $40,000$ records each one composed by the tweet content, its date, and the keyword used in the search.

Once the dataset was collected, we started the labeling phase, employing the expertise of a pool of 40 Italian sociology and political science students. Each student was assigned a set of 3000 tweets to be classified by means of a web application. The annotators performed an intensive training phase where an extensive set of tweets representing the three typologies have been proposed to them. Three different labels have been associated to each tweet, the first is the *political* label ($+1$: political, -1: not political), the second label *sentiment*, coded only for political tweets, is about the feeling and can assume the values $+1$ (positive or neutral) or -1 (negative), while the third is the binary label *general* ($+1$: general,-1: specific)[4].

The procedure was made so that each tweet was labelled by three different experts. This way we could increase the accuracy and the meaningfulness of the labelling process by removing tweets with disagreeing political labels. On the resulting set, we employed a majority voting approach for labelling the sentiment of the tweets. This choice was justified by the high level of agreement of the sentiment label. Indeed, by taking into account the Krippendorff's alpha coefficient[5], we obtained for the sentiment label $\alpha = 0.79$. In Table 1 we report the label distribution after applying the above procedure. Both political/not political classes and negative/positive classes within political tweets are quite balanced.

On the opposite, for the general label we obtain a low Krippendorff's alpha coefficient (0.41). It can be argued that "general speech" is a concept that could not unequivocally and objectively be defined for human beings, for this reason we chose to discard this label, and we decided to simplify this task by defining a rule-based approach described in Section 4.

The final dataset (TwitterTrainData) is composed by $28,340$ Italian labelled tweets on political and sentiment categories. To the best of our knowledge, it represents one of the biggest dataset containing tweets classified by experts.

[4]Two examples of Italian political tweets: "#ballaró: rappresentanti del nulla? Ci stiamo riprendendo i nostri diritti. State attenti, quello che avete visto é; solo l'inizio". Political, general, negative; "#agora in 30 minuti: la donna ventriloquo, l'autolesionismo della sinistra rappresentata da #vendola e #fassino". Political, specific, negative.

[5]Krippendorff's alpha coefficient [17] is a statistical measure of the agreement achieved when many evaluators code the same set of units of analysis in terms of the values of a variable.

3.2 Training Newspaper Data (NewsTrainData)

The adoption of TwitterTrainData in the training phase could present some drawbacks due to the limited period it spans. For example, some important features to achieve a good classification considering a narrow retrieval period might lose their relevance in a wider period. These drawbacks result in a limited generalization power of the model employed to classify the political tweets. To improve the generality, we built up an additional dataset (NewsTrainData) containing all the article titles of different Italian newspapers (*Repubblica*, *il Manifesto*, and *Libero*) so that they span the whole spectrum of the political points of view from the Right to the Left wing. More precisely, using the feed RSS history, we selected all the articles from January 1^{st}, 2012 to October 10^{th}, 2012 extracting the news title, and we employed the categorization proposed by the newspaper to associate a label to the title. If a news belongs to the political category proposed by the newspaper we set the label to $+1$, otherwise -1. The resulting NewsTrainData is composed by $17,388$ labelled newspaper titles, $10,670$ of which political (61%).

3.3 Italian Twitter Community Data

To obtain general results on the political disaffection we performed our analysis on a large sample of the Italian Twitter community. To achieve this goal, we randomly extracted $50,000$ Italian users, which posted at least one Italian tweet[6] in a fixed temporal range (October 10^{th} to October 30^{th}). Moreover, to extend our sample we selected for each user all its Italian followers, thus producing a set of $261,313$ users. Furthermore, we considered only the user profiles that have been created before April 4^{th} (obtaining $167,557$ users), to prevent the problem of the continuous growth of the Italian Twitter community, which could affect the quality of our political disaffection measure. Finally, we extracted all the tweets of each selected user, for the period of interest (April 4^{th}, 2012 to October 10^{th}, 2012), producing our final set composed by $35,882,423$ tweets (TweetCorpus). According to an estimate [7], it roughly represents more than 50% of the tweet volume posted in that period.

3.4 Public Opinion Surveys

Once detected all the tweets expressing political disaffection in TweetCorpus, we compared the resulting time series with the trend of some indicators extracted from public opinion surveys. The public opinion surveys have been collected by a global market research company (IPSOS) from April 11^{th}, 2012 to October 10^{th}, 2012. The sampling procedure consisted of a survey through CATI (computer-assisted telephone interview) of a representative sample of the Italian electorate. More precisely, almost every week, respondents were contacted with a quota sampling on fixed parameters (age, gender, education) using the technique of random digit dialing.

From public opinion surveys we extracted some indicators related to the phenomenon under investigation. An aspect of political disaffection that can be captured by public opinion survey is the attitude of political inefficacy. This attitude expresses the (subjective) sense of powerlessness of citizens in politics and the disbelief in the accountability of the political system and of all political parties. This definition led us to employ a measure of political inefficacy,

[6]To identify if a tweet is written in Italian we employ the Guess-Language library (https://code.google.com/p/guess-language/).

[7]http://daily.wired.it/news/internet/2012/09/26/numeri-twitter-italia.html

t_i	INEFFICACY	NO_VOTE
2012-04-11	14.86%	13.23%
2012-04-18	13.77%	14.53%
2012-05-02	22.20%	19.05%
2012-05-09	-	13.37%
2012-05-16	12.93%	13.34%
2012-05-23	16.31%	12.47%
2012-06-05	12.07%	13.31%
2012-06-06	11.03%	13.76%
2012-06-13	-	10.99%
2012-06-20	10.77%	13.08%
2012-06-26	6.91%	9.29%
2012-06-27	6.84%	13.09%
2012-07-04	-	11.88%
2012-07-11	7.87%	10.04%
2012-07-17	9.51%	13.64%
2012-07-18	6.00%	10.03%
2012-07-25	-	13.26%
2012-09-04	-	13.53%
2012-09-12	8.46%	11.22%
2012-09-19	-	12.75%
2012-09-25	9.44%	12.04%
2012-09-26	10.46%	12.74%
2012-10-03	-	12.87%
2012-10-10	11.76%	14.38%

Table 2: Public Opinion Surveys for INEFFICACY and NO_VOTE indicators.

that we called INEFFICACY, which represents the percentage of the respondents who expressed the minimum (equal to 1) propensity to vote (PTV, where the range is from 1-low to 10-high) for all political parties included in the surveys[8]. The PTV has been coded for each significative Italian party[9]. It is reasonable to assume that very low propensity to vote for any party is a reflection of the lack of accountability in the political system.

Together with political inefficacy we also used low intentions to vote as a proxy of political disaffection, which reflects citizens' sense of powerlessness (see [30]). We captured the intention to not vote at the next election using the NO_VOTE indicator. This indicator includes the percentage of survey respondents that declare to have extremely low intention to vote at the next elections[10] (see Table 2). In details, we considered the people that answered 1 (1 low - 10 high) to the question "How likely is it that you will vote at the next election?".

We want to emphasize that these two indicators refer to different aspects of political disaffection. INEFFICACY measures the scepticism of a citizen towards the whole political class and capture an attitude, which might or might not be associated to a behaviour. On the other hand, NO_VOTE can capture the potential behavioural consequences of political disaffection. Note that the propensity of this last behaviour is influenced by a variety of factors, including the proximity of political elections.

[8]The total sample consists in $38,537$ respondents (~ 2267 respondents per poll).
[9]PD (Partito Democratico), PDL (Popolo delle Libertà.), Lega Nord, IdV (Italia dei Valori), UDC (Unione di Centro), FLI (Futuro e Libertà.), SeL (Sinistra Ecologia Libertà.), and M5S (Movimento 5 Stelle).
[10]The total sample consists in $24,971$ respondents (~ 1040 respondents per poll).

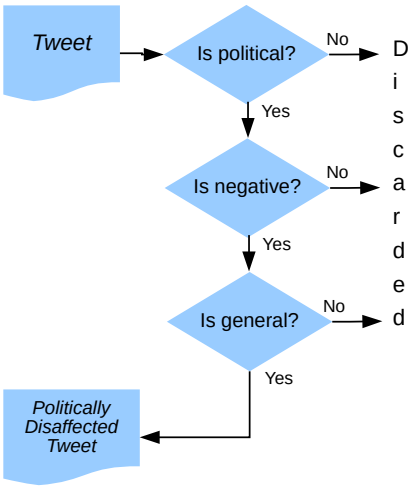


Figure 1: Classification chain employed.

4. CLASSIFICATION APPROACH FOR THE POLITICAL DISAFFECTION

Identifying political disaffection is a complex task even for human beings, so, in order to create a system for the detection of this attitude in tweets, we have to define it in a formal way. To that end, as described in Section 3, a political disaffection tweet has to match the following three criteria:

- **Political**: the tweet should regard politics.
- **Negative**: the sentiment of the tweet should be negative.
- **General**: the message have to regard politicians or parties in general. Tweets regarding only a political party or specific politician are not considered.

Since a classifier able to consider all this criteria at the same time can not be trained, we created a "chain" of classifiers as described in Figure 1. The relevant tweets after each step became the input for the next one. After every step the number of relevant tweets was less or equal to the number of relevant tweets after the previous step. The relevant tweets after the third step are eventually classified as *relevant* and all the other tweets are classified not relevant. Roughly speaking, we used TweetCorpus as input of the chain and we obtained a set of tweets denoting political disaffection (Tweet-Relevant) as output.
For the first step, we trained a classification algorithm using TwitterTrainData and NewsTrainData; the resulting classifier distinguished between political and non-political tweets. In the second step, the algorithm is trained with TwitterTrainData and the resulting classifier distinguishes between tweets with negative and non-negative sentiment. Please note that the TwitterTrainData collection is fixed, but the features are extracted in different ways depending on the classification step. The third and last step was performed by an ad-hoc classifier created with a rule-based approach to identify the general speech. Precisely, as noticed in Section 3, generality is a concept that could not unequivocally and objectively be defined for human beings, for this reason we opted to simplify this problem by employing a set of keywords identified by our experts to model

Keywords used for the general task	
politici	politicians
classe politica	political class
partiti	parties
deputati	members of parliament
senatori	members of senate
lo stato	the state
casta	clique

Table 3: The list of keywords selected by domain experts to model the general task in the political field.

this task (see Table 3). These keywords represent the most frequently used $\{1, 2\}$-gram of words that refer to the political class in general. Furthermore, to improve the generalization of this approach we employed DBpedia. This database allows performing queries and provides a simple and automatic way to capture the semantic behind words based on Wikipedia. By using this database we extracted all the Italian politicians with their political affiliation (left, centre, and right) and we substituted their names with their affiliations. We selected as general those tweets that contained the keywords identified by experts or at the same time all the possible political affiliations (left, centre, and right)[11].

In the next sections we summarize the feature extraction methodologies, and, subsequently, the results of different classification approaches.

4.1 Feature Extraction Approaches

The efficacy of textual classification crucially depends on how the textual data is transformed into numerical features. Nevertheless, identifying the best method for feature extraction is a non-trivial problem, and the results are usually task dependent. For these reasons, we separately managed the two supervised classification tasks: political topics and negative sentiment. Note that in political topics we employed both the tweets data (TwitterTrainData) and newspaper titles (NewsTrainData).

We compared different techniques for features extraction in order to find the most suitable for our problems: n-grams of characters, single words[12], $\{1, 2, 3\}$-grams of words, and we also applied more sophisticated approaches such string kernels [19].

For each of the techniques listed above, we computed: term frequency [20], boolean term presence [34], and term frequency-inverse document frequency (TF-IDF, [20]). Moreover, an important improvement was given by performing a stemming process and collapsing synonyms into a single feature. To perform this task we employed a freely available Italian synonyms dictionary[13].

The test has been performed for the political topic classification using a 4-fold cross-validation with an online linear classifier [14]. Our results showed that 5-grams of characters constitutes our best option, independently of the counting scheme.

Taking into account the negative sentiment task and replicating the experiments with the same methodology of the previous case, we noted that the feature characterized by the space-separated word tokenization achieves the overall best results, employing as counting scheme the term frequency. In order to make our classifier more robust, we removed in each tweet the object the sentiment is referred to (sentiment target) extracting from DBpedia a full and up-to-date set of Italian political parties, politicians, and political offices. Combining this data with the recognition of strings starting with "@" (Twitter user-names) we were able to remove the sentiment target from the tweets.

[11]The tweet: "#Bersani,#Berlusconi,#Monti sono tutti ladri" (#Bersani,#Berlusconi,#Monti are all thieves) becomes "left, centre, right are all thieves".

[12]Considering the space-separated words approach, we recognize as single word also emoticons and single punctuation marks such as "!". The URLs are also transformed in an unique token: ⟨link⟩).

[13]http://webs.racocatala.cat/llengua/it/sinonimi.htm

[14]Passive Aggressive, [10]

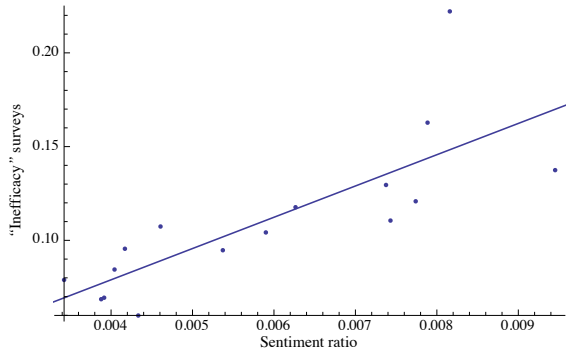


Figure 2: Linear fitting between the sentiment ratio (regressor) and the inefficacy indicator (response variable). The R-squared ($R^2 = 0.61$) and the significance value for the sentiment ratio coefficient ($\rho = 3 \cdot 10^{-4}$) stress the goodness of the fitted model.

4.2 Tested Classification Algorithms

To achieve our goal, we needed classifiers able to scale on huge corpus and possibly to be updated over time. Therefore, we especially focused our attention on online classifiers since they require only a single sweep on the data, making the classification process really fast with really good performances on the accuracy side.

We ran all the experiments on an ordinary workstation: Intel(R) Core(TM) i7-2600K CPU at 3.40GHz with 16Gb of RAM.

We tested four different online algorithms for classification[15] and one batch classification algorithm:

- **ALMA** [14]: is a fast classifier which try to approximate the maximal margin hyperplane between the two classes. We set the parameter p equals to 2.
- **OIPCAC** [26]: is a classification method that employs a modified approach to estimate the Fisher Subspace, which allows to manage classification tasks where the space dimensionality is bigger than, or comparable to, the cardinality of the training set, and to deal with unbalanced classes.
- **PASSIVE AGGRESSIVE** (PA, [10]): is a Perceptron-like method. In our experiments we tested only the binary classifier with different settings.
- **PEGASOS** [27]: is a well-known online Support Vector Machine (SVM) solver.
- **RANDOM FOREST** (RF, [6]): is an algorithm based on an ensemble of classification trees. Since the algorithm is widely used in machine learning challenges with good results, we will use it as yardstick in our comparison.

[15]Most of these algorithms are well known and have a MATLAB implementation available in DOGMA [23].

Classifier	Accuracy	F-Measure	Global time
ALMA	0.883 ± 0.014	0.886 ± 0.011	13.5 ± 1
PA	$\boldsymbol{0.889 \pm 0.012}$	$\boldsymbol{0.890 \pm 0.012}$	$\boldsymbol{10.62 \pm 0.1}$
PEGASOS	0.882 ± 0.010	0.883 ± 0.010	1103 ± 10
OIPCAC	$\mathbf{0.889 \pm 0.001}$	$\mathbf{0.891 \pm 0.010}$	$\mathbf{5911 \pm 52}$

Table 4: 10-fold results for political topic detection (in bold face the best results considering F-measure). In italic the classifier selected for the classification process. With "time", we intend the time employed for training and classification, in seconds. We were not able to conclude all the runs with RF due to its high requirement of resources. Note that the number of samples is $45,728$.

Classifier	Accuracy	F-Measure	Global time
ALMA	$\boldsymbol{0.703 \pm 0.029}$	$\boldsymbol{0.745 \pm 0.034}$	$\boldsymbol{0.82 \pm 0.28}$
PA	0.665 ± 0.064	0.705 ± 0.124	0.91 ± 10^{-3}
PEGASOS	0.691 ± 0.033	0.732 ± 0.045	76 ± 0.1
OIPCAC	0.714 ± 0.026	0.751 ± 0.024	121 ± 25
RF	$\mathbf{0.724 \pm 0.026}$	$\mathbf{0.776 \pm 0.027}$	$\mathbf{2173 \pm 48}$

Table 5: 10-fold results for negative sentiment detection (in bold face the best results considering F-measure). In italic the classifier selected for the classification process. With "time", we intend the time employed for training and classification, in seconds. Note that the number of samples is $12,476$.

We compared these algorithms on the two aforementioned tasks: political and negative. Note that we tested the online learning algorithms in a batch setting. In order to speed up the classification process we used only their one-sweep behaviour.
After an extensive tuning of the parameters, in Table 4 and in Table 5 we reported for each predictor its best performances in 10-fold cross validation on the political classification task and on the negative sentiment identification. In Table 4 the best result has been obtained by OIPCAC. The other classifiers achieved similar results, especially Passive Aggressive that further shows to be the fastest algorithm tested. In Table 5 the best result has been achieved by Random Forest, even though it had a very high running time. The variation between Table 4 and Table 5 highlighted that our sentiment classification task is more difficult w.r.t. the topic detection (political identification).

Considering the achieved results, in order to obtain a good trade off between accuracy and running times, we adopted the combination of PA and ALMA. In particular we used PA for the political/non-political classification and ALMA for the sentiment classification part. It is also possible to obtain comparable performances with different combinations of classifiers.

It is important to recall that since the general task (see Section 3) is difficult to be unequivocally modelled in all its aspect by a set of users, we simplified this problem by employing an approach based on a list of a few keywords selected by domain experts and by exploiting DBpedia as described above.

5. RESULTS

In this section we describe the time-series obtained employing the information extracted with the approach described in Section 4 and the relations between them and the public opinion surveys. Moreover, we summarize our methodology to identify the political news that produces the highest peaks of the generated time-series (breaking news).

To perform a correlation analysis with the INEFFICACY indicator taken from surveys, we employed the approach described in Section 4 to generate the set of tweets denoting political disaffection (TweetRelevant). Subsequently, taking into account each survey sampling date t_i (see Table 2), we generated three time-series computing the ratio between the number of political disaffection tweets and the number of political tweets by employing three time intervals:

1. from the date of the survey to 14 days before (Δ_1^{14});
2. from the day of the survey to 7 days before (Δ_1^{7});
3. from 7 days before the date of the survey to 14 before (Δ_7^{14}).

The same approach has been employed for the NO_VOTE indicator.

Table 6 shows the Pearson correlation index computed between the political disaffection tweet-series and the INEFFICACY time-series. The best result (0.79) represents a strong correlation value between INEFFICACY and the information extracted by our approach. Furthermore, it is important to stress that the best time interval is Δ_1^{14}. We can strengthen our result by analysing the dependency between the political disaffection ratio and the INEFFICACY indicator. As shown in Figure 2, we find a linear dependence between the variables expressed by line $y = 16.6x + 0.01$, i.e. an increase of the political inefficacy corresponds to an increase in the Twitter political disaffection. These results combined with the Twitter timeliness suggest that our approach would be able to capture change in disaffection more promptly than public opinion surveys as can be noticed in Figure 3.

Table 7 shows the Pearson correlation index computed between the political disaffection tweet-series and the NO_VOTE one. The best result (0.59) represents a medium correlation value but still relevant showing that there is some connection between the modelled political disaffection and the intention to not participate at the next election day.

5.1 Breaking News Identification

After verifying the correlation between INEFFICACY and the Twitter political disaffection, we employed the TweetRelevant data to empirically determine some of the possible causes that produce the variation in disbelief in politics and politicians, hypothesising that citizens' political inefficacy is affected by controversial political news reported daily in the media. To achieve this goal we identified the peaks of the time-series generated as the daily ratio between the number of political disaffection tweets and the number of political tweets, and we associated each peak to a news belonging to NewsTrainData.

More precisely, to identify the peaks we employed an approach similar to that proposed in [15], taking into account as peaks the points of the time series greater than $\mu+2\sigma$, where μ is the mean of the points of the time-series and σ is the standard deviation. However to improve the quality of our results we considered for each point a set of its neighbours[16] (instead of all the points) to estimate the local μs and σs. The qualitative results are shown in Figure 4.

[16]We use a temporal window of 10 days, 5 before and 5 after the point of the time-series taken into account.

Interval	ρ	95% Confidence Interval	P-Value ($\rho > 0$)
Δ_1^{14}	**0.7860**	**0.476-0.922**	**0.031%**
Δ_7^{14}	0.7749	0.454-0.917	0.042%
Δ_1^{7}	0.6880	0.310-0.878	0.226%

Table 6: Pearson correlation index achieved between Twitter political disaffection and INEFFICACY time-series (p-value and confidence interval are two-tailed).

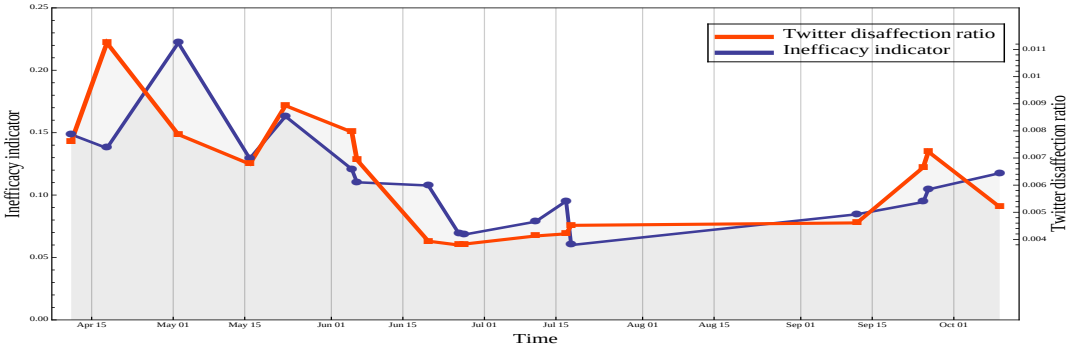


Figure 3: Twitter political disaffection time-series employing Δ_1^7 compared with the INEFFICACY indicator.

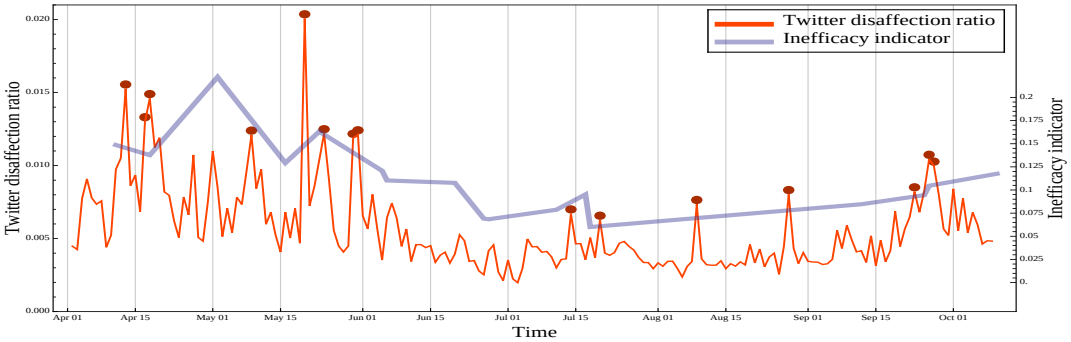


Figure 4: Political disaffection tweets day by day, with the selected peaks (highlighted by circles).

Interval	ρ	95% Confidence Interval	P-Value ($\rho > 0$)
Δ_1^{14}	**0.5920**	**0.248-0.803**	**0.231%**
Δ_7^{14}	0.5579	0.190-0.788	0.567%
Δ_1^7	0.4433	0.049-0.718	3.00%

Table 7: Pearson correlation index achieved between Twitter political disaffection and NO_VOTE time-series (p-value and confidence interval are two-tailed).

To associate each peak to a news, firstly we created an inverse document frequency (IDF) map by employing the words extracted from the corpus of the news included in NewsTrainData, and 1 million tweets randomly selected from the political subset (PTweetCorpus) of TweetCorpus (we employed the same classifier used for the political task described in Section 4 to identify the political tweets). Note that these weights reduce the relevance of the terms that are recurrent in many tweets. For each previously identified peak we create TD-IDF vectors for the tokenized news and tweets by employing the IDF map, thus obtaining, two vector sets for each day:

- $\mathcal{N}$ the vectors' set of the news;
- $\mathcal{T}$ the vectors' set of the tweets belonging to PTweetCorpus.

Subsequently we employed the cosine similarity between vectors to select the most correlated news with respect to the peak taken into account as follows:

$$\arg\max_{\boldsymbol{n}\in\mathcal{N}} \sum_{\boldsymbol{t}\in\mathcal{T}} \frac{\boldsymbol{n}\cdot\boldsymbol{t}}{\|\boldsymbol{n}\|\,\|\boldsymbol{t}\|}$$

The results achieved are summarized in Table 8 where we reported the news with the highest cosine similarity. Table 8 evidences that disaffection peaks correlate to a broad and diverse range of political news. The news spans from the discussion about economical Italian crisis and the contested labor reform to bribe scandals and the individual behaviors of leaders or members of a specific political party.

Finally, we qualitatively compared the news identified with this approach with the trending topics on Twitter related to the day of each peak and we could note that most of the news effectively corresponded to one of the political daily trend. However, for few peaks, NewsTrainData did not contain any news correlated with the majority of the tweets of that day. Looking at the Twitter trending topics, it can be argued that this happened whenever the political discussions on Twitter did not concern any facts reported in newspapers, but the discussions spontaneously grew in the Twitter community. A meaningful example concerns the trending topic #no2giugno: this movement asked for the suspension of the military parade of June the 2^{nd} (the Italian republic day), seen as a waste of resources, to use the money to rebuild the cities of Emilia (Italian region) after the earthquakes of 2012. This discussion generated two peaks (May 30^{th} and 31^{st}) that did not correlate with the traditional media news. A similar behaviour could be noted for the other two uncorrelated peaks.

2012-04-13	✓ La Lega prova a rifarsi un'immagine. Rinuncia agli ultimi rimborsi elettorali. Lega tries to clean up its image. It opts last electoral refunds out.
2012-04-17	✓ Lavoro, Monti pensa alla fiducia "I partiti approveranno la riforma". Labor, Monti thinks of a vote of confidence "Parties will enact reform".
2012-04-18	✓ Monti: niente crescita fino al 2013, disagio lavoro per meta famiglie. Monti: no economic growth until 2013, disadvantage for half of families.
2012-05-09	✓ Bersani: "Pd piú forte, Monti ci ascolti" Grillo: "Partiti morti". Crollo del Pdl. Bersani: "PD stronger, Monti listen to us" Grillo:"Parties are dead". PDL falls.
2012-05-20	✓ Grillo su Brindisi: strage di Stato, fa comodo a loro. Grillo about Brindisi: state massacre, it's convenient for them.
2012-05-24	× Grillo attacca: "Noi soldi non li vogliamo rinunceremo a rimborsi prossime politiche". Grillo bashes: "We don't need money, we opt last electoral refunds out".
2012-05-30	× Riforma Csm, il gelo di Monti cosí é fallito il piano di Catricalá. CSM reform, Monti's chill, Catricalá's project is doomed.
2012-05-31	× Spread, Monti resta preoccupato "Rischio contagio malgrado gli sforzi". Spread, Monti worried "Risk contagion despite the efforts".
2012-07-14	✓ Cicchitto: "Primarie sono inutili Berlusconi candidato premier". Cicchitto: "Primary election is useless, Berlusconi is the premier candidate".
2012-07-19	× Monti ora teme il crac della Sicilia. Now Monti is afraid of Sicily default.
2012-08-09	✓ Monti al WSJ: "Con Berlusconi spread a 1200" L'ira del Pdl. E votano contro il governo. Monti to WSJ: "Spread at 1200 with Berlusconi" PDL anger and vote against government.
2012-08-28	✓ Grillo a Bersani: "Io fascista? Tu sei un fallito d'accordo con la P2". Grillo to Bersani: "Am I Fascist? You're failed at one with P2".
2012-09-23	✓ Cosí si rubava alla Regione Lazio ecco le rivelazioni di Fiorito ai pm. Fiorito's admission to public prosecutor: "How we stole moneys from Lazio government".
2012-09-26	✓ Caso Sallusti, salta l'accordo con il giudice il direttore domani rischia il carcere. Sallusti's instance, legal agreement breaks, the lead director risks the jail.
2012-09-27	✓ Sallusti: "In Italia mancano le palle" Paolo Berlusconi respinge le dimissioni. Sallusti: "In Italy many wimps" Paolo Berlusconi rejects Sallusti's resignation.

Table 8: Each row represents the news with the highest cosine similarity identified by our approach. The symbol ✓ represents the identified news that also appears in the political Twitter trending topic of the day taken into account. As additional contextual information, Lega is an Italian party, Bersani, Cicchitto and Berlusconi are politicians, PD is the Italian Democratic Party, Monti is the Italian ex-premier, Sallusti and Paolo Berlusconi (Silvio Berlusconi's brother) are respectively the lead director and the editor of a newspaper, Grillo is a comic/politician, Fiorito is a regional councilman involved in bribe inquiry. CSM is the magistrates' internal board of supervisors. P2 is a secret society.

6. CONCLUSIONS AND FUTURE WORKS

In this work we analysed the well-known attitude of political disaffection by using Twitter data through machine learning techniques. While the majority of the research has concentrated on the investigation of Twitter users' political attitude in term of support to a specific candidate or party [31, 2, 29, 28] to our knowledge, no prior studies have analysed manifestation of political disaffection. To investigate this phenomenon we concentrated on the Italian case, which has been identified in the political science literature as one of the most extreme cases of disaffection. We found evidence of a diffuse and consistent political discourse that can be classified as political disaffection. In order to validate the quality of the time-series generated employing our approach, we compared our result with political disaffection as measured in public opinion surveys (low intentions to vote and low political efficacy). We found evidence of a strong correlation between the two time-series. Furthermore, we showed that important political news of Italian newspapers is often correlated with the highest peaks of the produced time-series. Taking into consideration the specificities of Twitter as a news medium (i.e. immediate diffusion of news and information), these results indicated that the data extracted from Twitter and from surveys are two reflections of a common underlying development that exhibits different temporal characteristics. This led us to suggest that Twitter data can be taken as a valid measurement of the fluctuating dimension of political disaffection.

This work showed that using Twitter to capture public opinion is more feasible than using it to predict the behavior of the public based on that opinion. Indeed, given the contradictory results of the electoral prediction based on Twitter data [21, 13], the different task of political disaffection seems to offer an interesting research topic for further investigations. This is suggested by the strong correlations between the time-series generated by employing the public opinion surveys and the Twitter data automatically extracted by our approach. Overall our results together with those presented in literature [4, 22] showed that for some phenomena the amount of Twitter discussion is a good measure of the diffusion of this phenomenon in society, despite the bias (such as age and geolocalization) of the Twitter population. The fact that we considered only people spontaneously writing on Twitter suggests that the resulting index is not a count of 'votes', but a measure of how much people are willing to spread these ideas in their lives (presumably not only using micro-blogs).

Moreover, Twitter's timeliness in relation to political events, with respect to traditional public opinion surveys, suggests that our method could be employed to perform a daily prediction of the citizen's political disaffection changes.

To further extend our approach and reach better results, we would like to improve our method to extract "generic speech" modelling this concept by means of ad-hoc ontologies [12]. Moreover we could enhance the classification accuracy with a proper selection of the tweets to be labelled by experts in an active-learning fashion (i.e. [9]), and we could improve the quality of the sentiment analysis by employing the top systems proposed in the SemEval 2013 competition. Furthermore, we could introduce the graph topology information in order to have a better understanding of the social component of this political phenomenon and the possibility to employ graph-based classifiers (i.e. [33], [16]).

Finally, since Twitter communications include a high percentage of ironic and sarcastic messages [28], another interesting improvement of our approach could be focused on the identification of these tweets as shown in [5].

7. REFERENCES

[1] A. G. Almond and S. Verba. *The Civic Culture. Political Attitudes and Democracy in Five Nations*. SAGE Publications, Inc., 1963.

[2] A. Bermingham and A. Smeaton. On using twitter to monitor political sentiment and predict election results. In *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP)*, 2011.

[3] J. Bollen and H. Mao. Twitter mood as a stock market predictor. *Computer*, 44(10):91–94, 2011.

[4] J. Bollen, A. Pepe, and H. Mao. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

[5] C. Bosco, V. Patti, and A. Bolioli. Developing corpora for sentiment analysis and opinion mining: the case of irony and senti-tut. *Intelligent Systems, IEEE*, 28(2):53–63, 2013.

[6] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[7] E. Cambria, B. Schuller, Y. Xia, and C. Havasi. New avenues in opinion mining and sentiment analysis. *Intelligent Systems, IEEE*, 28(2):15–21, 2013.

[8] E. Cambria, Y. Song, H. Wang, and N. Howard. Semantic multi-dimensional scaling for open-domain sentiment analysis. *Intelligent Systems, IEEE*, 2013.

[9] N. Cesa-Bianchi, C. Gentile, and F. Orabona. Robust bounds for classification via selective sampling. In *Proceedings of the Annual International Conference on Machine Learning (ICML)*, 2009.

[10] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

[11] G. Di Palma. Apathy and participation: Mass politics in western societies. *The Public Opinion Quarterly*, 36(2):297–299, 1972.

[12] L. M. Garshol. Metadata? Thesauri? Taxonomies? Topic maps! Making sense of it all. *Journal of Information Science*, 30(4):378–391, 2004.

[13] D. Gayo-Avello. I wanted to predict elections. *CoRR*, abs/1204.6441, 2012.

[14] C. Gentile. A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research*, 2:213–242, 2001.

[15] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *Proceedings of the International conference on World Wide Web (WWW)*, 2004.

[16] M. Herbster, M. Pontil, and S. R. Galeano. Fast prediction on a tree. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2008.

[17] K. H. Krippendorff. *Content Analysis: An Introduction to Its Methodology*. Sage Publications, Inc, 2nd edition, 2003.

[18] A. Livne, M. P. Simmons, E. Adar, and L. A. Adamic. The party is over here: Structure and content in the 2010 election. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

[19] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. J. C. H. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2:419–444, 2002.

[20] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[21] P. T. Metaxas, E. Mustafaraj, and D. Gayo-Avello. How (not) to predict elections. In *Proceedings of the Conference series on Social Computing (SOCIALCOM)*, 2011.

[22] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010.

[23] F. Orabona. *DOGMA: a MATLAB toolbox for Online Learning*, 2009. Software available at http://dogma.sourceforge.net.

[24] A.-M. Popescu and M. Pennacchiotti. "dancing with the stars", nba games, politics: An exploration of twitter users' response to events. In *Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

[25] R. D. Putnam. *Making democracy work: civic traditions in modern Italy*. Princeton University Press, 1993.

[26] A. Rozza, G. Lombardi, E. Casiraghi, and P. Campadelli. Novel fisher discriminant classifiers. *Pattern Recognition*, 45(10):3725–3737, 2012.

[27] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the Annual International Conference on Machine Learning (ICML)*, 2007.

[28] M. Skoric, N. Poor, P. Achananuparp, E.-P. Lim, and J. Jiang. Tweets and votes: A study of the 2011 singapore general election. In *Proceedings of the 45th Hawaii International Conference on System Sciences (HICSS)*, 2012.

[29] E. Tjong Kim Sang and J. Bos. Predicting the 2011 dutch senate election results with twitter. In *Proceedings of the Workshop on Semantic Analysis in Social Media (LASM)*, pages 53–60, Avignon, France, 2012.

[30] M. Torcal and J. Montero. *Political Disaffection in Contemporary Democracies: Social Capital, Institutions and Politics*. Routledge, 2006.

[31] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2010.

[32] N. Vallina-Rodriguez, S. Scellato, H. Haddadi, C. Forsell, J. Crowcroft, and C. Mascolo. Los twindignados: The rise of the indignados movement on twitter. In *Proceedings of the Conference series on Social Computing (SOCIALCOM)*, 2012.

[33] F. Vitale, N. Cesa-Bianchi, C. Gentile, and G. Zappella. See the tree through the lines: The shazoo algorithm. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, 2011.

[34] P. S. Yelena Mejova. Exploring feature definition and selection for sentiment classifiers. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media (ICWSM)*, 2011.

Enhancing Sentiment Extraction from Text by Means of Arguments *

Lucas Carstens
Imperial College London
South Kensington Campus
Huxley Building, Room 429
London SW7 2AZ, UK
lucas.carstens10@imperial.ac.uk

Francesca Toni
Imperial College London
South Kensington Campus
Huxley Building, Room 430
London SW7 2AZ, UK
ft@imperial.ac.uk

ABSTRACT

Sentiment Analysis is concerned with (1) differentiating opinionated text from factual text and, in the case of opinionated text, (2) determine its polarity. With this paper, we address problem (1) and present A-SVM (Argument enhanced Support Vector Machines), a multimodal system that focuses on the discrimination of opinionated text from non-opinionated text with the help of (i) Support Vector Machines (SVM) and (ii) arguments, acquired by means of a user feedback mechanism, and used to improve the SVM classifications. We have used a prototype to investigate the validity of approaching Sentiment Analysis in this multi faceted manner by comparing straightforward Machine Learning techniques with our multimodal system architecture. All evaluations were executed using a purpose-built corpus of annotated text and A-SVM's classification performance was compared to that of SVM. The classification of a test set of approximately 4,500 n-grams yielded an increase in classification precision of 5.6%.

Keywords

Sentiment Analysis, Support Vector Machines, Argumentation, User feedback, A-SVM

1. INTRODUCTION

Today, more than ever, the World Wide Web offers unprecedented opportunities to equally produce and consume data and information. At the same time, the immense pool of content emerging from a collaborative environment such as the Web carries significant implications when it comes to putting this content to use. Sentiment Analysis, or Opinion Mining, attempts to (1) differentiate opinionated text from factual text and, once text is deemed to be opinionated, (2) classify it as expressing a negative, neutral or positive opinion (see [18, 29] for an overview). The approach to Sentiment Analysis presented in this paper adds arguments, as understood in [23] and explained in section 2, to the probabilistic method of Support Vector Machines (SVM) [5] in order to distinguish opinionated from non-opinionated text and thus address problem (1) above. We have developed A-SVM, a *multimodal system* that classifies text according to its opinionatedness by means of combining SVM and arguments. We refer to A-SVM as multimodal because the system architecture incorporates concepts from both Machine Learning and Argumentation Theory. We have also developed a text corpus, consisting of approximately 13,000 annotated n-grams (each n-gram is part of a larger sentence) that are represented as feature vectors. In a preliminary evaluation, comparing A-SVM with a straightforward SVM classifier on our corpus, we achieved an increase in classification precision of 5.6%, from 78.1% to 83.7%, a 2.41% increase in recall and 4.05 points for the F1 measure.

Our approach can be seen as a novel way to integrate reasoning with rules (in the form of simple arguments) within probabilistic methods. Thus, we do not solely present a novel approach to Sentiment Analysis, but also investigate the merits of interweaving fields that have previously exhibited limited common ground. The main hypothesis supporting our approach is that combining substantially different methods of of dealing with written language computationally should allow an increase of general performance compared to applying each method in isolation. Our encouraging experimental results corroborate this hypothesis.

The paper is organised as follows: In section 2, we give an overview of the techniques used for the development and execution of A-SVM. In section 3 we introduce the corpus that forms the basis for text classification in our system, followed by A-SVM, the system itself, in section 4. In section 5 we present a twofold evaluation of the system, one quantitive, the other qualitative. The evaluation is followed by related work in section 6 and conclusions in section 7.

2. BACKGROUND

Our system makes use of SVM and arguments. Moreover, it is trained on a purpose built text corpus. In this section we first present the sources utilised in developing the corpus (section 2.1). We then go on to review SVM (section 2.2) and Argument Based Machine Learning (ABML) (section 2.3) from which we draw inspiration on how one may integrate arguments in Machine Learning.

*An extended version of this paper is available at `www.doc.ic.ac.uk/~lc1310`

WISDOM '13, August 11 2013, Chicago, USA.

2.1 Resources

The corpus we have developed during our research, described in detail in section 3, is built from a number of sources. We describe each of them below:

- The *MPQA corpus* [32] is a collection of 378 news articles, comprising around 10,000 sentences, each of which has been manually annotated with tags describing a number of subjectivity measures, as well as sources and objects of opinions within the text. The annotation scheme, proposed by Wiebe and colleagues, used to develop the MPQA corpus annotates the texts at word and phrase level. It thus applies relatively fine grained information about the articles annotated. The corpus annotations include "various properties involving intensity, significance, and type of attitude" [32], as well as records of the source of a statement and who a statement is directed towards. The complexity of the annotations is exacerbated by the way the annotations were contrived. All annotations were done manually and the annotators were given annotation guidelines that were rather loose and left the annotators with large room of choice in their annotation. The result is a very intricately annotated corpus. Since, for our purposes, the level of detail provided by the MPQA corpus exceeds our needs, we only use those annotations that define pieces of text as either opinionated or not opinionated.

- *SentiWordNet* [14] is a lexical resource that has been specifically designed for the purpose of aiding Sentiment Analysis, based on WordNet [20], but extended with lexical information about the sentiment of each synset contained in WordNet. A synset is a collection of words comprising all synonyms listed in WordNet for any particular word. The additional information provided by SentiWordNet, but not present in WordNet, comes in the form of three different values (positivity, objectivity and negativity) which sum to one and describe the orientation of sentiment.

- *TreeTagger* [31] is a publicly available Part-Of-Speech (POS) tagging system that uses decision trees for probabilistically selecting POS annotations. The aim of using decision trees in tagging POS is to allow taking into account the context in which the word or phrase that is tagged appears. The leaf nodes of the decision tree mark the actual decision on how to tag the word or phrase while the higher nodes give information about the surrounding words.

2.2 Support Vector Machines

Supervised Learning techniques have been the most prominently used techniques in Sentiment Analysis (see, for example, [16, 30]), with SVM yielding some of the most promising results (e.g. [24]). Here we briefly review SVM (see [5] for a more detailed introduction). Consider a linear model describing a binary classification problem such as the one of determining opinionatedness of a phrase. We can describe such a problem with a linear model of the form

$$y(x) = \mathbf{w}^T \phi(x) + b$$

Name	PaysReg.	Rich	HairColour	CreditAppr.
Mrs Brown	No	Yes	Blond	Yes
Mr Grey	No	No	Grey	No
Miss White	Yes	No	Blond	Yes

Table 1: Training Examples for ABCN2 learning algorithm

where $\phi(x)$ is a feature-space transformation of the data x (in our case a piece of text), $\mathbf{w}^T$ is a weight vector and b is a bias. Transforming data into feature space can yield linear separability of data that is not linearly separable in the original data space. A training data set consists of N input vectors $x_1, ..., x_N$, all of which have a class label $C \in \{1, -1\}$, and new data x is classified according to the sign of $y(x)$.

2.3 Argument-based Machine Learning

Argumentation (see [4] for an overview) can support the (dialectical) justification of conclusions. ABML [23] combines user feedback in the form of justified arguments and classical supervised machine learning (in the form of the CN2 algorithm [12]) to enhance performance. Usually supervised machine learning techniques take a preferably large number of training examples such as the ones used by Mozina and colleagues in [23], shown in table 1, and try to find a hypothesis that adequately explains the training examples and then correctly classifies new cases. In the example we have a number of parameters, e.g. *HairColour*, and a class label for each example, i.e. *CreditApproved*. Within the framework of ABML, some of these training examples have an associated argument explaining the reasoning behind why an example is classified the way it is. Consider again the example shown in table 1. The CN2 algorithm takes as input examples in the form of pairs (A, C), where A is an attribute-value vector, e.g. $(Name = MrsBrown, PaysRegularly = No, Rich = Yes, HairColour = Blond)$ and C is the class the example belongs to, e.g. $(CreditApproved = Yes)$. ABML accepts such examples, as well, but in addition is able to process examples of the form $(A, C, Arguments)$, where $Arguments$ is a set of arguments of one of the forms:

$$C \text{ because } Reasons \quad \text{or} \quad C \text{ despite } Reasons$$

where *Reasons* is a conjunction of attribute-value pairs such as

$$Arguments = \{C \text{ because } Rich = Yes, \\ C \text{ despite } PaysRegularly = No\}.$$

We will use arguments of analogous format but with n-grams as *Reasons* and classifications of (non-)opinionatedness as conclusions C.

3. THE TEXT CORPUS

We have developed a text corpus (available at `www.doc.ic.ac.uk/~lc1310`) of roughly 13,000 semi-automatically annotated n-grams, which was then used to train the SVM that classifies new text within A-SVM. The corpus was constructed using version 2.0 of the MPQA corpus, SentiWordNet and TreeTagger (see section 2.1), each of which contributed text, features annotating this text or both. Around

60% of the extracted n-grams were classified as opinionated and 40% as non-opinionated. Each n-gram in our corpus is associated with a vector of features describing certain characteristics of the n-gram. The maximum length of n-grams in the corpus is five words and all n-grams are extracted from the MPQA corpus. This value n$\leq$ 5 was chosen as a compromise between running into potential computational problems and hampering the ability to grasp the role that the context in which a word appears plays. The feature vector associated with an n-gram is comprised of up to nine features:

- one feature is the size of the n-gram;
- three features represent scores of positivity, neutrality and negativity extracted from the SentiWordNet lexicon;
- between one and five (given by n) additional features represent, for the words that appear in the n-gram, the words' lexical types and their basic form, which we obtain by applying TreeTagger to the n-gram.

In addition to the features, each annotated n-gram in our corpus contains a class label, $C \in \{+1, -1\}$, for the particular n-gram, identifying it as either *opinionated* ($C = +1$) or *non-opinionated* ($C = -1$). This class label is automatically extracted from the MPQA corpus, since the n-grams of the MPQA corpus are fully annotated with respect to opinionatedness. For example, our corpus contains the 2-gram *"refused to"*, annotated by

2 0.0 0.8125 0.1875
refused <VHZ >(refuse) to <TO >(to) +1

where n=2, positivity=0, neutrality=0.8125, negativity = 0.1875, followed by the POS tags and their basic forms and the classification +1, i.e. opinionated.

4. THE A-SVM SYSTEM

We present A-SVM, a multimodal system that tackles the discrimination of opinionated text from non-opinionated text with the help of SVM and a simple form of argumentation. A-SVM is, from a high-level perspective, comprised of a succession of input gathering, input conversion and input classification tasks. This succession is presented schematically in figure 1 and explained in detail throughout the remainder of this section. We refer to the *Activity* and *Data* nodes shown in figure 1 whenever the according part of the system is described below.

In a two-step classification process, SVM classifications (*"Preliminary classification"*) are improved via arguments, acquired by means of a user feedback mechanism, to obtain *"Final classification"*. The system has been trained on the corpus described above, but has been designed to analyse any textual input supplied by a user via a Graphical User interface (GUI) (*"GUI1: User input"*). We focus on the input conversion and input classification tasks. To describe the vital aspects of A-SVM we shall consider a single system execution from start to finish. Once initialised, the system prompts a GUI (*"GUI1: User input"*) to the user through which he or she provides and submits a piece of text, which we shall denote *TXT* subsequently. Let us assume the user has typed *TXT*: "*Despite the mounting pressure he has refused to bow*", taken from the MPQA corpus. *TXT* contains the 2-gram we have used as an example before, i.e. "*refused to*", and to describe the vital aspects of A-SVM we shall consider the n-gram (for n=4) "*has refused to bow*". This is an interesting n-gram because the word "*refused*" can be thought to be opinionated while we nevertheless argue that a clear opinionatedness of the 4-gram may be disputed.

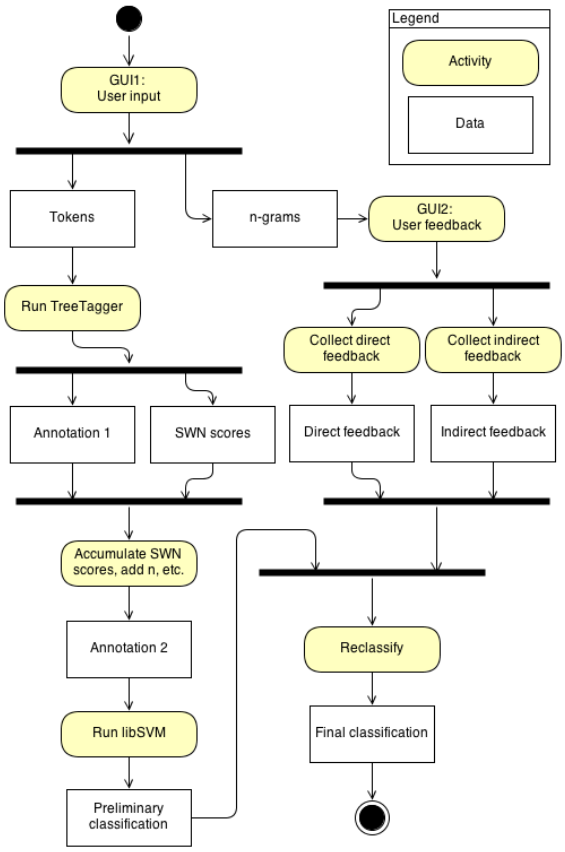


Figure 1: UML diagram showing a schematic view of A-SVM

Input conversion. Once the user has submitted the input (*TXT*) the first step in processing it is to break it up into single words (*"Tokens"*), each of which is annotated individually (*"Annotation 1"*). The entire annotation process takes place hidden from the user. The individual word annotations are then merged to form annotated n-grams (for all possible n-grams of up to five words in the given input). Each word is annotated with TreeTagger and with the scores extracted from the SentiWordNet lexicon, if the word occurs in the SentiWordNet lexicon, or with scores of zero, otherwise.

Resulting from this procedure are annotated single words

bearing five features each: scores for positivity, neutrality and negativity, the word's POS tag and its basic form. For example, the word "*has*" in our illustrative n-gram will be annotated as

0.0 1.0 0.0 has <VHZ >(have)

The next step is to construct annotated n-grams from the given input text, constructing them from the annotated single words. This step yields every possible annotated n-gram contained in the user input up to a length of five words. For any input text larger than three words, we obtain m=(n-2)*5 n-grams up to a size of five words. Each n-gram that is constructed is given one score for positivity, one for neutrality and one for negativity, obtained from the individual scores of the words in the n-gram, summed up and normalised by dividing by n. Lastly, each obtained n-gram is annotated with the size of the n-gram. For our example, the result of this procedure includes:

1 0.0 1.0 0.0 has <VHZ >(have)

2 0.0 0.8125 0.1875
has<VHZ >(have) refused<VVN>(refuse)

3 0.0 0.875 0.125
has<VHZ >(have) refused<VVN>(refuse) to<TO>(to)

4 0.0 0.78125 0.21875
has<VHZ >(have) refused<VVN>(refuse) to<TO>(to) bow<VV>(bow)

...

1 0.0 0.625 0.375 refused<VHZ >(refuse)

2 0.0 0.8125 0.1875
refused<VHZ >(refuse) to<TO>(to)

3 0.0 0.7083 0.2917
refused<VHZ >(refuse) to<TO>(to) bow<VV>(bow)

...

The final step before classifying TXT is a conversion of any non-numerical values in the annotations to numerical values, since SVM require strictly numerical inputs. This is done by simply assigning each possible POS tag a unique number between zero and one. This procedure produces what we refer to as *"Annotation 2"* in figure 1.

Preliminary classification. The SVM algorithm is applied to all annotated n-grams resulting from processing the input TXT as described above. This results in one class label, $C \in \{+1, -1\}$, per annotated n-gram, thus classifying each n-gram as being either opinionated (+1) or non-opinionated (-1) (*"Preliminary classification"*). The user is then presented with a second GUI (*"GUI2: User feedback"*) through which he or she is asked to provide the system with some (in the current version five) of their own classifications of n-grams, which are randomly selected from the original user input TXT. In addition to judging these n-grams as being opinionated or non-opinionated, the user gives a justification for this classification. This justification is given in the form of one argument per n-gram, in a syntax as described below.

Generating Arguments from user feedback. In giving arguments the user specifies the following:

1. The class label, $C \in \{+1, -1\}$, indicating whether the argument is in favour of the n-gram being *opinionated* or *non-opinionated*;
2. The direction of reasoning, i.e. *because* or *despite*;
3. The part of the n-gram that is *most* responsible for the user's judgement.

For example, a possible argument concerning the n-gram we have used before, "*has refused to bow*", may be

+1 (opinionated) because "refused to bow"

Here, *"refused to bow"* is the *Reason* for the conclusion $C = +1$ (see section 2.3). Thus, in this hypothetical scenario the user has judged the n-gram to be opinionated and identifies words two to four in the input as the reason for his or her judgement. In the same manner, the user provides four additional arguments by passing judgement upon four more randomly selected n-grams and providing reasons to support the decision. Below we refer to the five arguments given by the user as $ARGS$. The feedback is considered in two ways, as illustrated below.

Direct and indirect user feedback for reclassification. $ARGS$ consists of classifications for five n-grams in TXT and reasons for those classifications. Disregarding *Reasons* leaves a classification of the five n-grams. This is the direct feedback, used to overrule the classification of the SVM for those particular n-grams. We base this overruling on the choice to trust the manual classification (and thus the user) over the SVM classification. In order to gain information from the user's arguments that goes beyond simply reclassifying five n-grams, as done with the direct feedback, we construct what we call *indirect feedback*. This results in the re-classification of a subset of all n-grams that can be obtained from the original user input TXT. This subset consists of all n-grams that have the same structure, in terms of POS tags, as the n-grams the user classified through $ARGS$. Though this beckons further investigation, our preliminary evaluations hint that certain POS tag combinations may hold information about (non-)opinionatedness of text. Assuming this holds true, having the tags as indicators for a class means that it suffices that an n-gram in the text contains the same combination of POS tags as the n-gram that was used to construct the argument, while the n-gram does not have to be exactly the same.

Algorithmically, the indirect feedback works as follows: Each n-gram obtained from the original user input (TXT) is assigned a score, initially set to zero. Then, for each such n-gram, we check whether it has the same POS tag combination as one of the *Reasons* the user has provided in $ARGS$. If one such *Reason* exists in an argument with conclusion C this n-gram's score is either increased by one (if $C = +1$) or decreased by one (if $C = -1$).

At the end of this process, the original classification, i.e. *"Preliminary classification"*, is overwritten according to the sign of the score (to opinionated if the score is > 1 and to non-opinionated if it is < -1). If the value we obtain for an n-gram is between -1 and 1 we simply retain the original classification provided by the SVM. From this reclassification procedure we obtain our *"Final classification"*.

Algorithm 1 Pseudocode describing the final classification procedure

```
1: let N = {n_0, ..., n_4} be the n-grams for user feedback
2: let U = {u_0, ..., u_4} be the user feedback class labels
3: let A = {a_0, ..., a_4} be the feedback reasons
4: let m be the total number of n-grams constructed from
   the user input
5: let F = {f_0, ..., f_m} be feature vectors representing the
   original user input's POS tags
6: let L = {l_0, ..., l_m} be the class labels determined for
   F = {f_0, ..., f_m} by SVM
7: let V = {v_0 = 0, ...v_m = 0} be the indirect feedback
   values for F = {f_0, ..., f_m}
8: counter ← 0
9: while counter < m do
10:   for i = 0 to 4 do
11:     if f_counter == n_i then
12:       l_counter ← u_i
13:     else if a_i ∈ f_counter then
14:       if l_counter == +1 then
15:         v_counter + +
16:       else
17:         v_counter − −
18:       end if
19:     end if
20:   end for
21:   counter + +
22: end while
23: for j = 0 to m do
24:   if v_j > 1 then
25:     l_j ← +1
26:   else if v_j < −1 then
27:     l_j ← −1
28:   end if
29: end for
```

Note that some preprocessing of $ARGS$ is needed before executing the scoring algorithm. An argument of the form

- *opinionated because Reasons* is converted to the pair $< +1, POS >$,
- *opinionated despite Reasons* is converted to the pair $< -1, POS >$,
- *non-opinionated because Reasons* is converted to the pair $< -1, POS >$,
- *non-opinionated despite Reasons* is converted to the pair $< +1, POS >$,

where POS denotes the POS tag combination that is found in *Reasons* and POS is associated with a class label, $C \in \{+1, -1\}$, that is derived from the combination of opinionated (non-opinionated) and because (despite) as shown. For our example n-gram *"refused to bow"* we would attain the POS tag combination *VVN-TO-VV* and thus (1) is mapped to $< +1, VVN - TO - VV >$.

Final classification. This is the result of combining evidence from SVM and arguments provided by the user where the classification results of the SVM form the basis, with the arguments overruling the SVM classification whenever the evidence supplied by them is deemed strong enough. The calculation of the final classification is summarised in algorithm 1.

Line 11 checks whether the current feature vector has the equivalent POS feature values as one of the n-grams classified by the user feedback. If this is the case the class label l_m is overwritten with the user's classification (line 12). If this is not the case, we check whether the combination the user chose as being responsible for his or her choice of classification is part of the current n-gram's features (line 13). If this is the case, we either increase or decrease the $v_{counter}$, depending on the class label of the sub n-gram (lines 14 to 17). Depending on the value of $v_{counter}$, we ultimately determine our confidence in the part of user input being either opinionated or non-opinionated. After all $F = \{f_0, ..., f_m\}$ have been processed, all the n-grams' class labels whose indirect feedback value v_j surpass the threshold are changed accordingly (lines 23 to 27).

Classification output. Once we have obtained the final classification all that is left to do is presenting A-SVM's classifications to the user. An example output is shown in figure 2. Each word from the user input TXT, i.e. *"Although I like sunny weather,..."*, is assigned a score between *-1* and *+1*. This score is captured as follows. Each word of TXT will appear in more than one n-gram and will thus be classified more than once. Starting from zero we sum all classification results for each word; whenever a word appears in an n-gram that is classified as opinionated we increase the score by one; when the n-gram is classified as non-opinionated, we decrease it by one. The final sum is normalised, giving a value between -1 and 1. A graph (as in figure 2) is shown to the user to represent this *"vote of confidence"* of the system as to how sure it is that a word, in the context of the input sentence, is either opinionated or non-opinionated.

The closer a value is to $+1$, the more confident the system is that this word is opinionated, the closer we get to -1, the higher the system's confidence in this word's being non-opinionated. When the value tends to 0, the system has received conflicting evidence about this particular word's

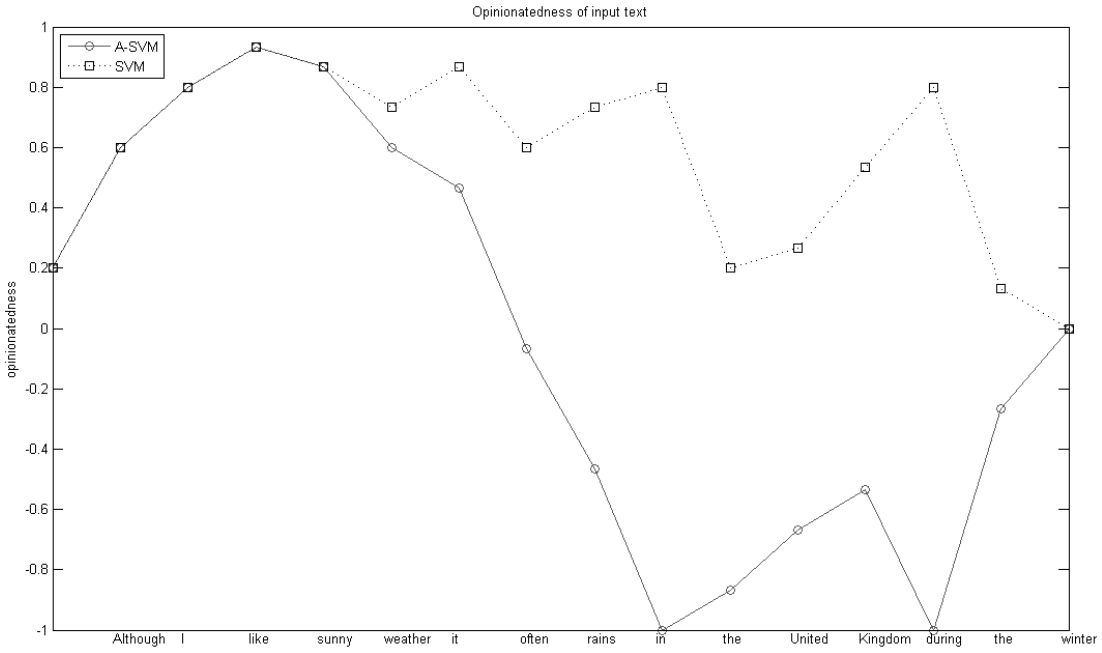


Figure 2: Classification results of an example user input for SVM and A-SVM

opinionatedness and is thus unable to identify a clear leaning. Figure 2 also shows the classification result (dotted line with boxes) using solely SVM, for the sake of comparison. As we can see, SVM arguably misclassifies most of the main clause ("it often rains...") whereas A-SVM seems to rectify this error.

5. SYSTEM EVALUATION

A number of supervised learning algorithms have been proposed to tackle different issues of Sentiment Analysis, some of which are compared in [30]. Pang and colleagues use three different classifiers for the task of classifying movie reviews at document level and compare the performance of Naive Bayes classifier, Maximum Entropy classifier and SVM. The results presented by Pang and colleagues show that, for the particular task of classifying movie reviews at document level, the best performance is achieved using SVM while the worst performing method is the Naive Bayes classifier. In line with these results we have focused on evaluating our system's performance using SVM.

The main question we are trying to answer is whether adapting a multimodal approach to Sentiment Analysis may prove valuable to the cause of furthering the development of systems. To do so, we put the classification performance of the developed system up against the classification performance of the SVM incorporated in the system, stripped bare of everything but the SVM itself. In addition to this comparison of classification performance, a user centred evaluation was conducted. This aimed at complementing the results of the quantitive evaluation with qualitative insights into the subjective impression users get from the system's performance.

Kernel	K. degree	gamma	K. coeff.	Cost	bias
linear	1	0.001	0	1	1
polynomial	to	to	to	to	to
RBF	10	100	1	60	10
sigmoid					

Table 2: parameter values tested for SVM training

5.1 SVM vs A-SVM

In order to evaluate the system's classification performance, the corpus of n-grams described in section 3 was split into a training set and a test set: 2/3 of the corpus were used to train the SVM classifier and 1/3 of the corpus was used as a test set. This division yielded a training corpus of roughly 8,500 n-grams and a test corpus of around 4,500 n-grams. Table 2 shows the range of parameter settings we have tried to achieve optimal performance on the SVM. We achieved the best performance using Radial Basis Function (RBF) kernels for which the kernel degree and kernel coefficient values have no influence on the performance. Setting the cost value $C = 1$ and $bias = 1$, as well, yielded the best results. Classifying the test set with just the SVM yielded precision of 78.1%, recall of 82.08% for opinionated n-grams and an F1 value ($F1 = 2 * \frac{precision*recall}{precision+recall}$) of 80.04 (see table 3). This constituted the baseline which the system was compared to. The classification results using the system as described above yielded precision of 83.7%, recall of 84.49% for opinionated n-grams and an F1 value of 84.09 (see table 3). The arguments needed to classify n-grams using A-SVM were provided manually. Using A-SVM thus yielded a performance increase of 5.6% in precision, 2.41% in recall and 4.05 points for the F1 measure.

	Precision	Recall	F1
SVM	78.1%	82.08%	80.04
A-SVM	83.7%	84.49%	84.09

Table 3: Evaluation results of SVM and A-SVM

5.2 Qualitative Analysis

Both the system and the SVM evaluation rely in their workings on qualitative judgements that have either been passed during the development of the corpus or during the classification process itself. Additionally, analysing text with regards to its sentiment often involves ambiguities that are owed not just to the context words and phrases are set in, but also the context a pieces of text may be written or read in or who it is written or read by. For this reason, complementing a quantitative evaluation such as the one described in the previous section with a qualitative evaluation has allowed us to gain a clearer understanding of whether or not the system is performing in a suitable manner. The qualitative evaluation was conducted with nine users and worked as follows: Each user was asked to use the system twice, once analysing a piece of text that was provided and once choosing his or her own text to be analysed. Having multiple users judge the same piece of text meant attaining easily comparable results while having the users choose their own text allowed an analysis of the system's robustness to unexpected input. After each of the two system executions the user was asked to judge a number of statements on a fivefold scale indicating how much the user agreed with the statement made. Figure 3 shows an excerpt from the questionnaire.

In addition to passing judgement on statements such as those shown in figure 3, the user was also asked to judge the general system's quality on a scale from one to five. This was asked to complement the more specific questions with a broader evaluation of the user's confidence in the system's performance and output. The average scores given to the system by this measure were 3.8 ($std = 0.414$) for the classification of the text that was given and at 3.667 ($std = 0.408$) for the user's own input. The remaining questions were aimed at judging both the perceived ease of use of the system and the perceived value of providing user feedback. For these questions we achieved scores similar to the overall classification scores.

6. RELATED WORK

Argumentation has, to the best of our knowledge, not been used to this date within the setting of Sentiment Analysis. It has however been applied in unison with Machine Learning techniques in other settings, e.g. [22, 23]. While we have focused on analysing any generic text input, many researchers have tended to focus their efforts on the analysis of customer reviews, e.g. [24, 28]. Most solutions, such as our own, have either been directed at extracting opinionated contents from text, e.g. [3, 7], or at identifying opinionated contents as being either positive or negative, e.g. [13, 30, 34]. Only some have proposed holistic systems that encompass the complete analysis of text, for example [33].

With rising interest in Sentiment Analysis, a number of text corpora, tailored to the needs and demands of Sentiment Analysis, have been developed. The most widely used have been the Multi-Perspective Question Answering (MPQA) corpus described in [32], which we have incorporated into our corpus, and the TREC (Text REtrieval Conference) blog tracks [19, 27]. The MPQA corpus annotates news articles and the TREC blog tracks use blog entries as source text.

[8] groups existing research into four categories: Keyword spotting, Lexical affinity, Statistical methods and Concept-based approaches. Lexical affinity assigns probabilistic values to words that determine how *affine* those words are to either other words [9] or an emotion. Our use of SentiWordNet scores as features of n-grams is similar to this concept. Concept-based approaches offer a deeper analysis of text, focusing on the semantics of it [2, 26]. This is realised through the use of web ontologies or semantic networks [6, 15]. We integrate some conceptual knowledge in our system via the user feedback.

Statistical methods have by far received the most attention in the Sentiment Analysis community and numerous algorithms have been used to approach the issue. Though supervised learning techniques (SVM as well as others) have been among the more popular ones researchers have also proposed applications using both Unsupervised Learning techniques, e.g. [3, 10, 34], and, more recently, Reinforcement Learning techniques, e.g. [7, 11, 33]. In [16] Kim and Hovy present a system for determining sentiment polarity which uses the concept of seed words to construct a classification model. A small amount of such seed words is collected which are either unambiguously positive or negative and annotated accordingly. This list is then iteratively expanded using synonymy and antonymy relations in WordNet. Such an approach requires significantly less effort than manually constructing a text corpus. Since we are classifying n-grams rather than words, however, a seed word approach was not feasible. In [7] Breck and colleagues use Conditional Random Fields (CRF) to distinguish opinionated text from non opinionated content based on the MPQA corpus. They also collect a number of features describing the text, among which are syntactical features determined by a POS tagging system, and have a CRF algorithm subsequently classify expressions according to those features. Choi and colleagues [11] apply CRF not to identify opinions but rather to find sources of opinions. Using various features that determine syntactic, semantic, and orthographic lexical characteristics of text, they train a CRF to identify both sources of direct and indirect opinions.

7. CONCLUSION

We have described A-SVM, a novel multimodal system for discriminating opinionated text from non-opinionated text that combines standard SVM and arguments from user feedback. We have trained and evaluated our system on a novel corpus and have shown experimentally that A-SVM outperforms standard SVM on the given corpus. We have additionally conducted a small qualitative analysis of A-SVM, the results of which show a rather favourable judgement of the users who have tried the system and, despite the relatively small sample size of nine users, we are confident that these results reflect the performance of A-SVM rather well. The results corroborate our hypothesis in section 1 that combining substantially different methods of of dealing with written language computationally should allow an increase of

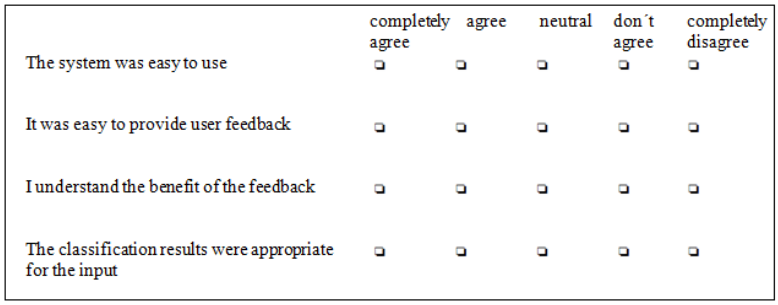


Figure 3: Excerpt from the questionnaire filled out by the evaluation participants

general performance compared to applying each method in isolation. Though we have both trained and evaluated A-SVM on a specific collection of news articles (as our corpus is based on the MPQA corpus [32]) during the qualitative evaluation the users were asked to submit text of their choice to A-SVM. The quality of the resulting classifications hints that with some effort we may be able to achieve a certain degree of domain independence. Though we have focused on using SVM for our classification, future efforts should be directed towards evaluating different algorithms, such as Maximum Entropy classifiers. In addition to testing different algorithms we need to conduct more extensive evaluations for each of them, including cross validation and more varied parameter settings.

The main issue left unaddressed in our work is the determination of opinion polarity. Numerous approaches have been presented to address this issue, e.g. [10, 30, 34]. Some of these efforts have focused on sentiment polarity exclusively, while others cover the full spectrum from determining opinionatedness to the summarisation of classification results. In light of the proposed system, there are two basic ways how polarity determination may be integrated into the system:

1. Add additional passes of binary decisions to the classification process
2. Develop a classifier that is able to make decisions on multiple classes at once

While the first alternative may prove to be the simpler solution, it may bring with it excessive computational demands and thus prove to be an unsustainable quick fix. In contrast to this solution, the second approach to integrating opinion polarity into the system's concerns would require fundamental changes to the system architecture but may prove beneficial with regards to computational efficiency. In addition to this rather central challenge, future improvements upon A-SVM may include tasks such as enhancing and extending the corpus, using different corpora to train the system and validate its performance or integrating further learning processes which continuously update a data base that contains not just the original corpus, but all past user inputs, as well.

By achieving measurable improvements upon a unimodal pattern recognition procedure we believe to have made a strong case for the potential benefits of going beyond pattern recognition algorithms in Sentiment Analysis. As has been suggested by some, e.g. Pat Langley in [17], it may prove to be necessary in the future to shift focus away from sheer statistical analysis to more complex tasks as envisioned when Machine Learning was still in its infancy. As Langley states:

> *"I do not believe that we should abandon any of the computational advances that have occurred in the [past] 25 years [...]. Each has been a valuable contribution to our understanding of learning. However, I think it is equally important that we not abandon the many insights revealed during the field's early period, which remain as valid today as when they initially came to light. The challenge for machine learning is to recover the discipline's original breadth of vision [...]."*

We argue that the concept of achieving text classification by combining established Machine Learning algorithms with Argumentation techniques allows us to make a step to achieving the above mentioned breadth of vision. Though subject to further investigation, this may hold true not just for basic binary classification of opinionatedness, but also multi-class classification for Sentiment Analysis as well as other NLP problems, such as Word Sense Disambiguation [25], Paraphrasing [1] or Argumentation Mining [21].

8. REFERENCES

[1] I. Androutsopoulos and P. Malakasiotis. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38(1):135–187, 2010.

[2] A. Balahur, J. M. Hermida, and A. Montoyo. Detecting implicit expressions of sentiment in text based on commonsense knowledge. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 53–60. Association for Computational Linguistics, 2011.

[3] M. Baroni and S. Vegnaduzzo. Identifying subjective adjectives through web-based mutual information. In *Proceedings of KONVENS*, volume 4, pages 17–24. Citeseer, 2004.

[4] T. J. M. Bench-Capon and P. E. Dunne. Argumentation in artificial intelligence. *Artif. Intell.*, 171(10-15):619–641, 2007.

[5] C. Bishop. *Pattern recognition and machine learning*, volume 4. Springer New York, 2006.

[6] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[7] E. Breck, Y. Choi, and C. Cardie. Identifying expressions of opinion in context. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 2683–2688. Morgan Kaufmann Publishers Inc., 2007.

[8] E. Cambria, B. Schuller, Y. Q. Xia, and C. Havasi. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, 28(2):15–21, 2013.

[9] D. Carmel, E. Farchi, Y. Petruschka, and A. Soffer. Automatic query refinement using lexical affinities with maximal information gain. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 283–290. ACM, 2002.

[10] Y. Choi and C. Cardie. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 793–801. Association for Computational Linguistics, 2008.

[11] Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 355–362. Association for Computational Linguistics, 2005.

[12] P. Clark and T. Niblett. The cn2 induction algorithm. *Machine learning*, 3(4):261–283, 1989.

[13] K. Dave, S. Lawrence, and D. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM, 2003.

[14] A. Esuli and F. Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422. Citeseer, 2006.

[15] M. Grassi, E. Cambria, A. Hussain, and F. Piazza. Sentic web: A new paradigm for managing social media affective information. *Cognitive Computation*, 3(3):480–489, 2011.

[16] S. Kim and E. Hovy. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, 2004.

[17] P. Langley. The changing science of machine learning. *Machine Learning*, 82:275–279, 2011.

[18] B. Liu. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*, 2, 2010.

[19] C. Macdonald, I. Ounis, and I. Soboroff. Overview of the TREC 2007 blog track. In *Proceedings of TREC 2007*, 2007.

[20] G. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[21] R. Mochales and M. Moens. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22, 2011.

[22] M. Možina, J. Žabkar, T. Bench-Capon, and I. Bratko. Argument based machine learning applied to law. *Artificial Intelligence and Law*, 13(1):53–73, 2005.

[23] M. Možina, J. Zabkar, and I. Bratko. Argument based machine learning. *Artificial Intelligence*, 171(10-15):922–937, 2007.

[24] T. Mullen and N. Collier. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of EMNLP*, volume 4, pages 412–418, 2004.

[25] R. Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10, 2009.

[26] D. J. Olsher. Full spectrum opinion mining: Integrating domain, syntactic and lexical knowledge. In *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*, pages 693–700. IEEE, 2012.

[27] I. Ounis, M. De Rijke, C. Macdonald, G. Mishne, and I. Soboroff. Overview of the TREC-2006 blog track. In *Proceedings of TREC*, volume 6. Citeseer, 2006.

[28] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics, 2004.

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

[30] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[31] H. Schmid. Probabilistic part-of-speech tagging using decision trees, 1994.

[32] J. Wiebe, T. Wilson, and C. Cardie. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2):165–210, 2005.

[33] T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. OpinionFinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, pages 34–35. Association for Computational Linguistics, 2005.

[34] H. Yu and V. Hatzivassiloglou. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing-Volume 10*, pages 129–136. Association for Computational Linguistics, 2003.

Evaluation of an Algorithm for Aspect-Based Opinion Mining Using a Lexicon-Based Approach

Florian Wogenstein
University of Applied Sciences
Alfons-Goppel-Platz 1
95028 Hof, Germany
fwogenstein@iisys.de

Johannes Drescher
University of Applied Sciences
Alfons-Goppel-Platz 1
95028 Hof, Germany
jdrescher@iisys.de

Dirk Reinel
University of Applied Sciences
Alfons-Goppel-Platz 1
95028 Hof, Germany
dreinel@iisys.de

Sven Rill
Goethe University Frankfurt
University of Applied Sciences
Alfons-Goppel-Platz 1
95028 Hof, Germany
srill@iisys.de

Jörg Scheidt
University of Applied Sciences
Alfons-Goppel-Platz 1
95028 Hof, Germany
jscheidt@iisys.de

ABSTRACT

In this paper, we present a study of aspect-based opinion mining using a lexicon-based approach. We use a phrase-based opinion lexicon for the German language to investigate, how good strong positive and strong negative expressions of opinions, concerning products and services in the insurance domain, can be detected. We perform experiments on hand-tagged statements expressing opinions retrieved from the Ciao platform. The initial corpus contained about 14,000 sentences from 1,600 reviews. For both, positive and negative statements, more than 100 sentences were tagged. We show, that the algorithm can reach an accuracy of 62.2% for positive, but only 14.8% for negative utterances of opinions. We examine the cases, in which the opinion could not correctly be detected or in which the linking between the opinion statement and the aspect fails. Especially, the large gap in accuracy between positive and negative utterances is analysed.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering*; I.2.7 [**Natural Language Processing**]: Text analysis

General Terms

Algorithms

Keywords

Aspect-based opinion mining, lexicon based approach, opinion lexicon

WISDOM '13, August 11 2013, Chicago, USA.

1. INTRODUCTION

The need of techniques for an automatic analysis of textual data has raised as the amount of text data has increased in the Web 2.0. Beside other issues, the extraction of opinions from text data becomes more and more important.

Opinion mining can be performed on several levels. The document-level opinion classification is the less granular approach. The aspect-based opinion mining, aiming to find out opinions uttered about aspects or features of entities, is the most fine-grained approach.

An example for entities are technical devices like mobile phones with aspects such as the display or the battery. Another example for an entity is a company with its products or services as aspects. Even human beings can be regarded as entities about which opinions are uttered. In this case, aspects are the attributes describing them or, e.g., their skills on specific things.

We describe a quite simple algorithm used to extract aspects and opinion bearing phrases, retrieve opinion values from an opinion lexicon and map the phrases to the aspects. The opinion lexicon lists opinion phrases and their opinion values for the German language. Thus, as the phrases directly include negation words and valence shifters, the task of opinion composition is much easier compared with other approaches, where the opinion values have to be derived from single word opinion lexicons.

We test our algorithm using contributions in the area of insurances retrieved from a German review platform. Thus, we deal with opinion utterances concerning insurance companies, their products and services.

2. RELATED WORK

During the last decade, a lot of research work has been done in the area of opinion mining.

Overviews of the different topics of opinion mining or sentiment detection have been given in [22] and recently in [12] as well as in [6].

The aspect-based opinion mining can be performed using supervised learning techniques, several groups have discussed this approach, see [2, 14, 40]. However, the supervised learning approach is highly dependent on the training data used.

As the training data normally covers one specific domain, the trained system is not easily applicable to a wider range of application domains.

The lexicon-based approach has been applied and discussed in several publications, see for example [9]. Here, opinion lexicons are used to determine the opinion values for statements expressing opinions. Most opinion lexicons list words with their opinion values, negation words and valence shifters (intensifiers and reducers). Thus, the opinion values of phrases have to be composited by the several values for the basic words. The procedure of opinion composition is discussed in [7, 18, 23, 25, 36].

Opinion lexicons exist for several languages. For the English language, e.g., widely used resources are SentiWordNet [1, 10, 11], WordNet-Affect [34] and Semantic Orientations of Words [35], all three generated using the WordNet®[24] lexical database, the Subjectivity Lexicon [41], SenticNet [5] and lists of positive and negative opinion words provided by [21]. Also for the German language, opinion lexicons exist, namely a polarity lexicon described in [8], listing about 8,000 opinion words with their opinion values, GermanPolarityClues [39] with more than 10,000 opinion words and SentiWS [30] with about 3,500 words. All these resources only include opinion values for single words.

The generic approach to derive the opinion lexicon used for this work has been described in [32], the generation of the list for the German language, called Sentiment Phrase List (SePL)[1] has been discussed in [31]. It includes 2,833 phrases with a length of up to five words each. However, it just contains adjective- and noun-based phrases, but it does not yet include any verbs or verb-based phrases.

Also opinion lexicons for other languages exist, e.g., for Spanish [4].

Applications of opinion mining are widely discussed. Online reviews are used for several purposes, examples are classification [28, 38], summarization [27] and the evaluation of the helpfulness of reviews [26]. Another emerging field is the detection of review spam, see [15, 16].

3. THE ALGORITHM

The algorithm to perform the aspect-based opinion mining is done in several steps which are described in the Sections 3.1 to 3.4. Figure 1 depicts the whole process.

At the beginning, some preprocessing steps are necessary. Afterwards, the aspects, which are relevant for the domain under consideration, are extracted. The next step is the detection and classification of opinion bearing phrases. An opinion lexicon for the German language is used to classify these phrases into strong positive and strong negative expressions of opinions. At the end, the opinion bearing phrases are linked to the associated aspects.

The study does not aim to obtain results for specific insurance companies. Thus, in our examples, the names of the insurances are masked as ABC and XYZ in the following.

3.1 Preprocessing

In a first step, sentences are separated using the Apache OpenNLP[2] Sentence Detector. Afterwards, both, the Apache OpenNLP Tokenizer and the Apache OpenNLP Part-Of-Speech Tagger, are applied to separate words and to assign

[1]http://www.opinion-mining.org/

[2]http://opennlp.apache.org/

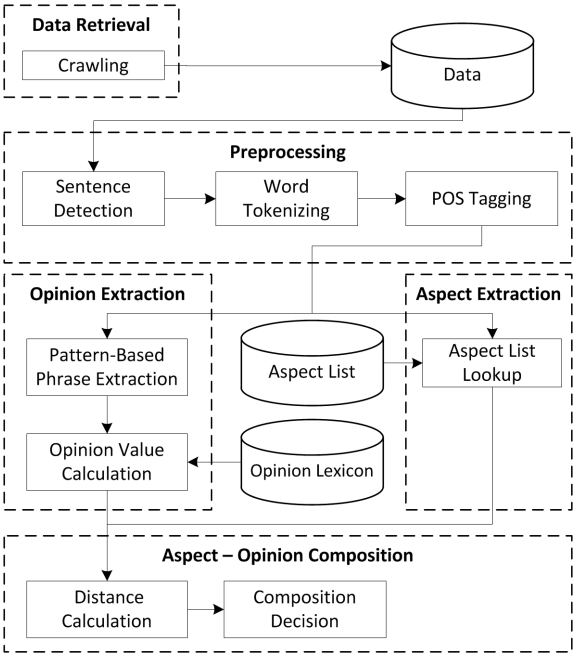


Figure 1: Overview of the algorithm.

the correct POS-tags to them. Therefore, the model corresponding files for the German language are used. For the POS-tagger the maximum entropy model, trained using the TIGER treebank [3], was used. The POS-tags are given in the Stuttgart-Tübingen Tagset (STTS) [33] systematics.

Sometimes, the POS-tagging is erroneous. Especially in cases, where an adjective is the first word in a sentence and therefore written with a capital letter, it occasionally gets tagged as a noun. To deal with this problem, all words at the beginning of a sentence tagged as nouns are tagged again with the capital letter replaced by a small letter. In addition, the Stanford POS-tagger [37] is used. If both algorithms tag the word as an adjective, the POS-tag is changed from noun to adjective.

3.2 Aspect Definition and Extraction

As we have chosen the insurance domain for our experiments, only entities and aspects from this domain have to be extracted from the text. Therefore, an entity and aspect model was produced in order to organize the entities (insurances) and connected aspects.

This model was generated in a semi-automated way. In a first step, entities and aspects were collected manually in a base list. Afterwards, this list was extended using the community-generated German synonym lexicon OpenThesaurus [3]. At the end, the list was normalized by lemmatizing all words using the German Morphology-Lexicon[4], which is based on Morphy[5] [19].

In the following, we do not distinguish between entities and aspects any more, as they are treated exactly in the same

[3]http://www.openthesaurus.de/

[4]http://www.danielnaber.de/morphologie/

[5]http://www.wolfganglezius.de/doku.php?id=cl:morphy

way. Table 1 shows some entities and aspect groups as well as examples for synonyms and sub-aspects.

Entities	**Synonyms**
ABC-Versicherung - *ABC Insurance*	ABC - *ABC* ABC-Versicherung - *ABC-insurance*
XYZ AG - *XYZ plc*	XYZ - *XYZ* XYZ-Versicherung, XYZ Versicherung - *XYZ insurance* XYZ-AG - *XYZ-plc*
Aspect Groups	**Examples for Sub-Aspects**
Produkt - *product*	KFZ-V. - *automobile insurance* Sach-V. - *property insurance* Unfall-V. - *accident insurance*
Kosten - *costs*	Versicherungsbeitrag - *insurance premium* Gebühr - *fee*, Rabatt - *discount*
Service - *service*	Servicemitarbeiter - *service staff* Kundenservice - *customer service* Beratung - *consulting*
Konditionen - *conditions*	Vertrag - *contract* Angebot - *offer* Tarif - *tarif*

Table 1: Some entities and aspect groups in the insurance domain.

The extraction of the aspects is carried out as a simple search. Due to the fact, that some aspects span over more words, the longest possible aspect phrase is taken, e.g., as "Dienst" - *'service'* and "Öffentlicher Dienst" - *'public service'* both are aspects, the latter will be taken as the aspect for the opinion mapping.
Within the extraction of the aspects from the texts, the same lemmatizer as used for the generation of the list is applied.

3.3 Handling of Opinion Bearing Phrases

In our approach, an opinion bearing phrase consists of at least one opinion bearing adjective or noun and up to four additional parts like intensifiers, negation words, adverbs or other adjectives. Examples for opinion bearing phrases are "brilliant" - *'brilliant'*, "sehr gut" - *'very good'*, "nicht wirklich besonders gut" - *'not really especially good'* or "Schrott" - *'junk'*. Two steps have to be performed to obtain the opinion expressed. In a first step, the opinion bearing phrases are extracted based on the patterns mentioned above. Afterwards, opinion values are looked up from the opinion lexicon and converted into three classes (strong negative, strong positive, other).
As the opinion lexicon we chose the phrased-based opinion list for the German language SePL, described in [31].

3.3.1 Extraction of the Phrases

For extracting the opinion bearing phrases, the same patterns as for the generation of the opinion lexicon were used [31]. Due to the fact, that in the opinion lexicon all words are lemmatized, a lemmatizing of the phrases was necessary to obtain the opinion values from the lexicon. Some minor changes had to be applied to the procedure in order to lemmatize all words in the right way and especially to suppress the lemmatizing for comparative and superlative forms. Otherwise, phrases like "beste" - *'best'* would have been lemmatized to "gut" - *'good'* which is not wanted as the former expresses a stronger opinion compared to the latter.

3.3.2 Application of the Opinion Lexicon

The opinion lexicon includes opinion values for both, single opinion bearing words and phrases with a length of up to five words.
While using the opinion lexicon, in most of the cases it is possible to obtain an opinion value for a given opinion bearing phrase extracted from the text directly. An example for this is the phrase "wirklich sehr gut" - *'really very good'*. However, sometimes phrases are missing in the opinion lexicon. If the phrase just consists of one word, there is no possibility to determine an opinion value. If the phrase consists of more than one word, the phrase is shortened by one word and another lookup in the opinion lexicon is performed. If the shortened phrase occurs in the list, the corresponding opinion value is taken in all cases where the shortening does not cut off negation words. If, for example, the phrase "sehr sehr gut" - *'very very good'* occurs but is not present in the opinion lexicon, the phrase is shortened to "sehr gut" - *'very good'*. On the other hand, if the phrase "nicht ausgesprochen gut" - *'not exceptionally good'* is found and its opinion value can not retrieved from the opinion lexicon, it will not be shortened as omitting the negation word would change the tonality drastically.
At the end, each opinion phrase is categorized as strong positive, if the opinion value is greater than 0.67 and as strong negative, if the opinion value found is smaller than -0.67. At this point, we assume that opinion values can be classified into three equal-sized intervals (negative, neutral and positive) and that both, the positive and negative class, can be divided again into subclasses representing strong and weak polarities.
Table 2 lists examples of strong positive and strong negative opinion bearing phrases with their opinion values. We distinguish between adjective- and noun-based phrases. All examples are directly taken from the test data described in Section 4. Also shown is one example of an opinion phrase which has an opinion value of less than 0.67, thus being not regarded as strong positive ("freundlich" - *'kind'*).

Adjective-Based Phrases	***OV***	***|sp/sn***
großartig - *great*	0.94	sp
sehr günstig - *very low priced*	0.89	sp
kompetent - *competent*	0.77	sp
freundlich - *kind*	0.58	–
mies - *lousy*	-0.71	sn
nur schlecht - *just bad*	-0.88	sn
Noun-Based Phrases	***OV***	***|sp/sn***
Herz - *heart*	0.83	sp
Mogelpackung - *bluff package*	-0.70	sn
Frechheit - *impertinence*	-0.91	sn

Table 2: Some words and phrases with their opinion values and the categorization into strong positive (sp) and strong negative (sn).

3.4 Distance-Based Linking

The linking of the opinion phrases to the aspects is done using a distance-based approach applied on the sentence-level. All strong positive or negative opinion phrases are linked to the next aspect found in a sentence according to the word position.

A simple example is the sentence "Ich bin sehr enttäuscht von dem Service." - *'I am very disappointed of the service.'*. The opinion phrase "sehr enttäuscht" - *'very disappointed'*, having an opinion value of -0.78 and thus being strong negative, is linked to "Service" - *'service'*. The result is an opinion tuple giving the opinion phrase, the tonality (sn = strong negative, sp = strong positive) and the aspect itself. In the example, the opinion tuple is "<sehr enttäuscht |sn| Service>" - *'<very disappointed |sn| service>'*.

As the opinion holder in a forum in most of the cases is just the writer of a post and the insurance is mostly given directly or in the title of the forum thread, an opinion quintuple defined in [13, 20] giving the entity, the aspect, the opinion, the opinion holder and the time could be constructed in most of the cases.

If more aspects than opinion phrases are found, the opinion phrase is linked to both aspects, e.g., "Die Mitarbeiter und der Service sind sehr gut!" - *'The employees and the service are very good!'*. This results in the tuples "<sehr gut |sp| Mitarbeiter>" - *'<very good |sp| employees>'* and "<sehr gut |sp| Service>" - *'<very good |sp| service>'*.

If more opinion phrases than aspects are found, e.g., one aspect and two opinion phrases, only the nearest phrase is linked to the aspect.

4. EXPERIMENTS

4.1 Test Data

For our experiments we chose the domain 'automobile insurances'. Data from Ciao[6], which provides a review platform for a wide variety of products and services, were retrieved using a tailored crawler. To split the sentences, the Apache OpenNLP Sentence Detector was used again. The total corpus consist of about 14,000 sentences extracted from about 1,600 reviews concerning about 120 insurances. Comments to the posts were not considered. The length of the sentences used for this study was restricted to be lower than 200 characters to avoid mistakes done by the sentence tokenizer. Errors occurred for example, if a sentence delimiter is used in an improper way, e.g., "Der Service ist lausig!!!" - *'The service is lousy!!!'*, or just if the whitespace after the sentence delimiter is missing.

After these preselection steps, approximately 12,000 sentences remained.

4.2 Manual Classification of Sentences

To be able to quantify the accuracy of our algorithm, sentences were classified manually. The task to perform was to tag strong opinions expressed about aspects of insurances. The tagging was done sequentially by two persons, in the following called annotators A and B, not involved in the project and not aware of the algorithm used for the opinion mining later on. A possible result of the tagging was, e.g., "Der Service ist miserablel, aber die Mitarbeiter sind sehr nett. <Service | sn>, <Mitarbeiter | sp>" - *'The service is lousy, but the employees are very kind. <service | sn>, <employees | sp>'*.

The sentences to be tagged were preselected randomly from the test data corpus. The only selection criterion was the presence of at least one aspect per sentence. The tagging was performed until each more than 100 strong positive and strong negative expressions of opinions were found.

At the end, 221 sentences with 234 aspects remained. A strong positive attitude was tagged for 119 of them, in 115 cases the author expressed a strong negative opinion about the aspect.

As already mentioned, the tagging persons were not informed about the algorithm, especially they did not know that the opinions are searched according to opinion bearing phrases. Thus, also sentences, where the opinions were expressed in an indirect way or using idioms, were accepted. An example would be "ABC-Versicherung - vergiss es!" - *'ABC-Insurance - forget it!'*.

To quantify the agreement of the tagging persons, we calculated Cohen's kappa coefficient, getting a value of $\kappa = 0.821$ with a p-value lower than 0.001. Thus, the agreement of the tagging results was very good for the two persons doing the manual tagging. For the calculation of the accuracies, the tagging results of annotator A were taken as the reference.

5. RESULTS AND DISCUSSION

5.1 Experimental Results

For both subsets, strong positive and strong negative expressions of opinions, we calculated the accuracy. To be counted as correct, both, the detection of the tonality (strong positive or strong negative opinion uttered) and the link to the aspect, had to be correct. Using the algorithm described above, we found that 74 out of the 119 positive statements were recognized correctly, while only 17 out of the 115 negative statements could be detected in the right way.

Thus, we reach an accuracy of 62.2% for the strong positive and only an accuracy of 14.8% for the strong negative expressions of opinions.

In the following, we give examples of statements correctly detected as being positive or negative, see Section 5.1.1.

The main purpose of our work is the investigation of the cases, in which the opinion utterance can not be detected correctly. We manually analysed the error sources statement by statement. In the following Sections (5.1.2 to 5.1.10), we discuss the reasons for the failure of our algorithm. In principle, more than one error could be present for one aspect in one sentence. For example, in the sentence "Ich hasse diese bescheuerte Versicherung" - *'I hate this dumb insurance'* it could be the case that the word "bescheuert" - *'dumb'* is not in the opinion lexicon and in addition, the writer uses a verb-based phrase. However, these cases occur very rarely so statements like this are counted only once in the determination of the error sources.

Before discussing the error sources in detail, we have to say that we found no case in which a positive statement was assessed as being negative or vice versa. Such an error typically would occur, if a negation is overseen or treated in the wrong way. As we said before, we used an opinion lexicon which directly includes negation and valence shifting in the phrases, leading to the fact that there is no need for a special treatment of negation and valence shifting. We have to admit, that cases of a wrong treatment of negations can not be ruled out completely. However, these cases seem to be rare enough not to play a role in this study.

[6]http://ciao.de/

5.1.1 Statements Correctly Detected

In the following, we give some examples for statements which were recognized correctly in an ascending order of sentence complexity.

- "Toller Service!" - *'Great service!'*
- "Ich bin sehr enttäuscht von der ABC Versicherung." - *'I am very disappointed of the ABC-Insurance.'*
- "Aus meinem Bekanntenkreis, durch den ich auch letztendlich bei der ABC gelandet bin, habe ich nur Positives gehört." - *'From my friends, which brought me to the ABC, I heard only good things about it.'*
- "Sicherlich ist die ABC nicht die billigste Versicherung auf dem Markt, doch mit Abstand eine der besten, was Service, Kundendienst und Leistungen angeht." - *'For sure, the ABC is not the cheapest insurance, but by far one of the best according to the service, the customer satisfaction and the insurance benefits.'*

5.1.2 Opinions Expressed via Opinion Bearing Verbs

The opinion lexicon used just contains opinion phrases based on polar adjectives and nouns. Thus, opinions expressed using verbs can not be detected. In 19 (16%) of the positive and 43 (37.4%) of the negative statements authors used verbs to express their opinions. An example is "Insgesamt kann ich die ABC-Versicherung empfehlen" - *'All in all I recommend the ABC-insurance'*.

5.1.3 Indirect Expression of Opinions

In 4 cases (3.5%) of negative expressions, the opinion was uttered in an indirect way. An example for such an indirect statement is "Hier eine kleine Geschichte über die XYZ, meine Ex-Versicherung." - *'Here a little story about the XYZ, my ex-insurance.'*. For the sentences expressing a positive opinion, this case did not occur.

5.1.4 Opinions Uttered with Idiomatic Expressions

In 4 (3.4%) of the positive and 16 (13.9%) of the negative statements, the opinions were uttered using an idiomatic expression. An example for this is "Finger weg von der ABC Versicherung!" - *'ABC insurance - Hands off!'*.

5.1.5 Wrong Links of the Opinions to the Aspects

In our approach, we allow for more than one opinion bearing phrase and also for more than one aspect in a single sentence. Thus, the links of the phrases to the aspects can be wrong. In our sample, this error occurred in 5 cases (4.2%) of the positive and also 5 cases (4.3%) of the negative statements. An example for this is "Die Servicezeiten sind hier nicht so toll, die Kundenbetreung hingegen ist einmalig gut" - *'The service hours are not so good, the customer service, on the other hand, is brilliant'*. "nicht so toll" - *'not so good'* has a smaller distance to "Kundenbetreuung" - *'customer service'* so it is linked to the wrong aspect.

5.1.6 Phrases Missing in the Opinion Lexicon

In some cases, the opinion was expressed using an adjective or noun phrase, nevertheless, it could not be resolved as the phrase was missing in the opinion lexicon. An example for such a statement is the word "Inkompetenz" - *'incompetence'*, which was missing in the opinion list used. This error occurred in 6 (5%) of the positive and 19 (16.5%) of the negative statements.

5.1.7 Wrong Opinion Values in the Opinion Lexicon

Sometimes, the opinion phrase is included in the opinion list, but the opinion value is below the threshold for strong positive and strong negative words. This error occurred for 1 positive utterance (0.8%) and for 2 negative ones (1.7%). The associated opinion phrases were "schnell" - *'fast'* with an opinion value (OV) of 0.089, "Angst" - *'fear'* (OV = 0.252) and "mangelhaft" - *'insufficient'* (OV = −0.661).

5.1.8 Irony and Sarcasm

In 2 cases (1.7%), negative opinions were expressed using irony, an example was "Ich dachte, die XYZ ist ihr Geld wert, aber wenn es darauf ankommt, kann man sich mal wieder auf die 'Versicherung' verlassen." - *'I thought, the XYZ is worth its money, but when it comes to the crunch, again you can count on the "insurance".'*. For the positive statements, this error did not occur.

5.1.9 Spelling Mistakes and Specialities

Text sources show a wide range of the grade of correctness concerning grammar and spelling. Especially, text data retrieved from Web 2.0 platforms are often written in a very 'creative' way.
In 6 cases (5.0%) for positive and also 6 cases (5.2%) for negative statements, spelling mistakes lead to the fact that the phrases could not be recognized correctly. An example was "TOLLE Versicherung!" - *'GREAT insurance!'*. Here, the POS-tagger does not recognize "TOLLE" - *'GREAT'* as an adjective and therefore, the pattern-based phrase recognition for the application of the opinion lexicon fails.

5.1.10 Comparisions

In 4 cases (3.4%) of the positive and 1 case (0.9%) of the negative statements, the opinion values could not determined correctly due to comparisons used. An example is "Jetzt bin ich bei einer Direktversicherung, die ist um einiges günstiger als die XYZ" - *'Now I am with a direct insurance which is significantly cheaper then the XYZ'*.

5.2 Summary

We want to summarize the results of the investigation of the errors occurring during the opinion phrase extraction and the linking of the phrases to the aspects.
Our algorithm is based on an opinion lexicon including only adjective- and noun-based phrases, so up to now it is not capable of dealing with verb-based phrases. Nevertheless, we also allowed opinions expressed via verb-based phrases as we wanted to find out the fraction of the usage of verbs in expressions of opinions.
Thus, we calculate the accuracy in two ways, once including verb-based phrases (a) and once excluding them for the study (b).
Table 3 gives an overview of the frequency of the several error sources for positive statements, Table 4 for negative utterances of opinions.

We can say, that for some categories of problems a solution will be possible. It is clear, that for both, positive and negative statements, the inclusion of verb-based phrases is essential as the lack of these is the main error source in both cases.
The main error sources apart from the missing verb phrases are improper links of the phrases to the aspects due to the

Statements	Number	Percentage
Total - strong positive (a)	119	100.0%
Correctly recognized (a)	74	62.2%
Total - strong positive (b)	100	100.0%
Correctly recognized (b)	74	74.0%
Error Source	**Number**	**Percentage**
Verb-based phrases	19	16.0%
Indirect opinion expressions	0	0.0%
Idiomatic expressions	4	3.4%
Wrong links	5	4.2%
Phrases missing	6	5.0%
Wrong opinion value	1	0.8%
Irony / Sarcasm	0	0.0%
Spelling mistakes	6	5.0%
Comparisons	4	3.4%

Table 3: Statistical summary for strong positive statements, (a) including verb-based phrases and (b) excluding them.

Statements	Number	Percentage
Total - strong negative (a)	115	100.0%
Correctly recognized (a)	17	14.8%
Total - strong negative (b)	72	100.0%
Correctly recognized (b)	17	23.6%
Error Source	**Number**	**Percentage**
Verb-based phrases	43	37.4%
Indirect opinion expressions	4	3.5%
Idiomatic expressions	16	13.9%
Wrong links	5	4.3%
Phrases missing	19	16.5%
Wrong opinion values	2	1.7%
Irony / Sarcasm	2	1.7%
Spelling mistakes	6	5.2%
Comparison	1	0.9%

Table 4: Statistical summary for strong negative statements, (a) including verb-based phrases and (b) excluding them.

simple distance assignment, missing phrases in the opinion list and wrong POS-tags due to spelling mistakes. Especially for negative utterances, another main error source is the usage of idiomatic expressions.
In the following, we want to discuss possible solutions for the error sources listed above:

- Errors due to the wrong links between aspects and opinion phrases can occur in cases where two opinions are uttered about two aspects using a main and a subordinate clause, for an example see Section 5.1.5. A solution for this problem could be the use of a sentence parser to split main and sub- clauses in order to apply the distance-based linking of aspects to opinion phrases only within the splitted (sub-)clauses. We performed first experiments using the Stanford Sentence Parser[7] to split up the sentences using the German PCFG model[17, 29]. Results are looking quite promising, but it can not yet be told, in how many cases the problem of wrong links can really be solved.

- The problem of missing phrases in the opinion list could be solved to a certain extend by expanding the opinion list. Up to now, the list was constructed only using Amazon reviews. These reviews almost are written to evaluate products, only a few of them contain statements concerning services. This leads to the fact that a big part of the vocabulary used to express opinions about services is missing in the list. Thus, other review platforms, especially sources providing reviews concerning services, could be used to enrich the opinion list. For example the Ciao platform, used for this study only as a source for the statements, could be used to find additional and domain specific opinion bearing words. Moreover, for concrete applications also a manual enhancement could be feasible.

[7]http://nlp.stanford.edu/software/lex-parser.shtml

- Errors occurring due to misspelling can be one of the most serious problems when applying opinion mining on data retrieved from Web 2.0 platforms. In our study, the main effect of misspellings were wrong results in the POS-tagging step (see Section 5.1.9). The application of spell checking and correction can help to solve this problem. In some cases, were the problem is just a misuse of capital letters (see the example in Section 5.1.9), one could try to repeat the POS-tagging after replacing the capital letters by small ones. However, it is not yet clear, in how many of the cases this really solves the problem.

- Idiomatic expressions are widely used for statements expressing opinions, especially for negative utterances, see Section 5.1.4. Thus, the opinion lexicon will be extended with these idiomatic expressions.

5.3 Shortcomings and Future Work

The accuracy obtained with our approach looks a little bit sobering, especially for negative utterances of opinions. Here, we want to discuss several limitations of our work and point out the way of our future work. One main source for an improvement is the absence of verb-based phrases in the opinion lexicon used for this study. As the tonality of many verbs is highly domain-dependent, a special treatment of verbs is necessary. In our view, a special taxonomy of opinion words has to be set up for the domain under investigation to treat a high percentage of the verbs in the correct way.
At the moment, we do not resolve coreferences. This leads to the fact that sentences, in which an aspect is not directly stated but is coreferenced, e.g., using a pronoun, are not selected into our sample. An example for this would be "Die Mitarbeiter der ABC sind sehr kompetent. Sie könnten aber schneller reagieren." - *'The staff of the ABC are very competent. But they could be a little bit faster.'*. The second sentence would not pass our preselection as we only take sentences in which at least one aspect is included, see Section 4.2. Furthermore, there might be the same problem within one sentence. For example "Die Mitarbeiter sind sehr

freundlich, aber sie sind nicht kompetent." - *'The employees are very kind, but they are not competent.'* the latter opinion phrase would not be assigned to the Aspect "Mitarbeiter" - *'employees'*.
In this work, we only regard statements expressing strong opinions. We have to admit that the detection of weak positive or negative utterances is quite a lot more difficult. Thus, the accuracy drops down as soon as one includes less 'drastic' statements.
In the next steps, we are planning to include verb-based phrases into the opinion lexicon and address the problems described in Sections 5.1.4, 5.1.5 and 5.1.6, where we see the biggest chance for a significant improvement of our approach. Another task to be done is to compare our method with a machine learning approach. Furthermore, we want to apply the aspect-based opinion mining to other domains and to texts retrieved from different data sources to learn more about possible sources of problems.

6. CONCLUSIONS

In this paper, we presented our approach for the application and evaluation of aspect-based opinion mining.
We looked at strong positive and strong negative statements, written in German, about insurances, their products and services with a quite simple algorithm.
It uses a phrase-based opinion lexicon for the German language and a simple distance-based algorithm for linking the opinion phrases to the aspects. Thus, it does not require any training and is applicable to many different domains and text sources.
We showed that it is possible to reach an accuracy of 62.2% for strong positive statements, but only of 14.8% for negative ones.
The purpose of the work was the analysis of the error sources. The main shortcoming of our approach are the missing verb-based phrases in the opinion list, being responsible for about 16% of not correctly detected positive statements and for about 37% of not correctly detected negative ones.
For negative phrases, two other main error sources exist, namely missing phrases in the opinion list and the use of idiomatic expressions, which are missing in the opinion list, too.
Also other errors occur for both, positive and negative statements.
Our impression is that authors of negative statements use a wider range of possibilities for expressing their opinions, leading to the fact that the correct treatment of negative utterances of opinions is more challenging compared to the detection of positive statements.
However, we think that some of the error sources could be eliminated or at least diminished by improvements of the algorithms used.

7. ACKNOWLEDGEMENTS

The authors would like to thank all members of the Institute of Information Systems (iisys) of the University of Applied Sciences Hof for many helpful discussions and especially R. Göbel for his great efforts as to the foundation of the institute.
The project is supported by the Bavarian Ministry of Economic Affairs, Infrastructure, Transport and Technology.
The Institute of Information Systems is supported by the Foundation of Upper Franconia and by the State of Bavaria.

8. REFERENCES

[1] S. Baccianella, A. Esuli, and F. Sebastiani. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 2200–2204, 2010.

[2] E. Boiy and M.-F. Moens. A machine learning approach to sentiment analysis in multilingual web texts. *Information Retrieval*, 12(5):526–558, 2009.

[3] S. Brants, S. Dipper, S. Hansen, W. Lezius, and G. Smith. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, pages 24–41, 2002.

[4] J. Brooke, M. Tofiloski, and M. Taboada. Cross-Linguistic Sentiment Analysis: From English to Spanish. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 50–54, 2009.

[5] E. Cambria, C. Havasi, and A. Hussain. SenticNet 2: A Semantic and Affective Resource for Opinion Mining and Sentiment Analysis. In *Proceedings of the 25th International Florida Artificial Intelligence Research Society Conference*, pages 202–207, 2012.

[6] E. Cambria, B. Schuller, Y. Xia, and C. Havasi. New Avenues in Opinion Mining and Sentiment Analysis. *IEEE Intelligent Systems*, 28(2):15–21, 2013.

[7] Y. Choi and C. Cardie. Learning with Compositional Semantics as Structural Inference for Subsentential Sentiment Analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 793–801, 2008.

[8] S. Clematide and M. Klenner. Evaluation and Extension of a Polarity Lexicon for German. In *Proceedings of the 1st Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 7–13, 2010.

[9] X. Ding, B. Liu, and P. S. Yu. A Holistic Lexicon-Based Approach to Opinion Mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 231–240, 2008.

[10] A. Esuli and F. Sebastiani. Determining Term Subjectivity and Term Orientation for Opinion Mining. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 193–200, 2006.

[11] A. Esuli and F. Sebastiani. SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 417–422, 2006.

[12] R. Feldman. Techniques and Applications for Sentiment Analysis. *Communications of the ACM*, 56(4):82–89, 2013.

[13] M. Hu and B. Liu. Mining and Summarizing Customer Reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, 2004.

[14] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao. Target-dependent Twitter Sentiment Classification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human*

Language Technologies - Volume 1, pages 151–160, 2011.

[15] N. Jindal and B. Liu. Review Spam Detection. In *Proceedings of the 16th International World Wide Web Conference*, pages 1189–1190, 2007.

[16] N. Jindal and B. Liu. Opinion Spam and Analysis. In *Proceedings of the 1st ACM International Conference on Web Search and Data Mining*, pages 219–230, 2008.

[17] D. Klein and C. D. Manning. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 423–430, 2003.

[18] M. Klenner, S. Petrakis, and A. Fahrni. Robust Compositional Polarity Classification. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 180–184, 2009.

[19] W. Lezius, R. Rapp, and M. Wettler. A Freely Available Morphological Analyzer, Disambiguator and Context Sensitive Lemmatizer for German. In *Proceedings of the 17th international conference on Computational linguistics - Volume 2*, pages 743–748, 1998.

[20] B. Liu. Sentiment Analysis and Subjectivity. In N. Indurkhya and F. Damerau, editors, *Handbook of Natural Language Processing, Second Edition*. 2010.

[21] B. Liu, M. Hu, and J. Cheng. Opinion Observer: Analyzing and Comparing Opinions on the Web. In *Proceedings of the 14th International World Wide Web Conference*, pages 342–351, 2005.

[22] B. Liu and L. Zhang. A Survey of Opinion Mining and Sentiment Analysis. In C. C. Aggarwal and C. Zhai, editors, *Mining Text Data*, pages 415–463. Springer US, 2012.

[23] J. Liu and S. Seneff. Review Sentiment Scoring via a Parse-and-Paraphrase Paradigm. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 161–169, 2009.

[24] G. A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38:39–41, 1995.

[25] K. Moilanen and S. Pulman. Sentiment Composition. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, 2007.

[26] S. M. Mudambi and D. Schuff. What Makes a Helpful Online Review? A Study of Customer Reviews on Amazon.com. *MIS Quarterly*, 34:185–200, 2010.

[27] J.-C. Na, T. T. Thet, C. S. G. Khoo, and W. Y. M. Kyaing. Visual Sentiment Summarization of Movie Reviews. In *Proceedings of the 13th International Conference on Asia-pacific Digital Libraries*, pages 277–287, 2011.

[28] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: Sentiment Classification using Machine Learning Techniques. In *Proceedings of the Conference on Empitical Methods in Natural Language Processing*, pages 79–86, 2002.

[29] A. N. Rafferty and C. D. Manning. Parsing Three German Treebanks: Lexicalized and Unlexicalized Baselines. In *Proceedings of the Workshop on Parsing German*, pages 40–46, 2008.

[30] R. Remus, U. Quasthoff, and G. Heyer. SentiWS – a Publicly Available German-language Resource for Sentiment Analysis. In *Proceedings of the 7th International Conference on Language Ressources and Evaluation*, pages 1168–1171, 2010.

[31] S. Rill, S. Adolph, J. Drescher, D. Reinel, J. Scheidt, O. Schütz, F. Wogenstein, R. V. Zicari, and N. Korfiatis. A Phrase-Based Opinion List for the German Language. In *Proceedings of KONVENS 2012*, pages 305–313, 2012.

[32] S. Rill, J. Drescher, D. Reinel, J. Scheidt, O. Schütz, F. Wogenstein, and D. Simon. A Generic Approach to Generate Opinion Lists of Phrases for Opinion Mining Applications. In *Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM)*, 2012.

[33] A. Schiller and C. Thielen. Ein kleines und erweitertes Tagset fürs Deutsche. In *Tagungsberichte des Arbeitstreffens "Lexikon + Text", 17./18. Februar 1994, Schloß Hohentübingen*, Lexicographica Series Maior, pages 193–203. Niemeyer, Tübingen, 1995.

[34] C. Strapparava and A. Valitutti. WordNet-Affect: an Affective Extension of WordNet. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 1083–1086, 2004.

[35] H. Takamura, T. Inui, and M. Okumura. Extracting Semantic Orientations of Words using Spin Model. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 133–140, 2005.

[36] T. T. Thet, J.-C. Na, and C. S. G. Khoo. Aspect-based sentiment analysis of movie reviews on discussion boards. *Journal of Information Science*, 36:823–848, 2010.

[37] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, pages 173–180, 2003.

[38] P. D. Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424, 2002.

[39] U. Waltinger. GermanPolarityClues: A Lexical Resource for German Sentiment Analysis. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pages 1638–1642, 2010.

[40] W. Wei and J. A. Gulla. Sentiment Learning on Product Reviews via Sentiment Ontology Tree. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 404–413, 2010.

[41] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of the Human Language Technology Conference*, pages 347–354, 2005.

Commonsense-Based Topic Modeling

Dheeraj Rajagopal
NUS Temasek Laboratories
Cognitive Science Programme
117411, Singapore
dheeraj@nus.edu.sg

Daniel Olsher
NUS Temasek Laboratories
Cognitive Science Programme
117411, Singapore
olsher@nus.edu.sg

Erik Cambria
NUS Temasek Laboratories
Cognitive Science Programme
117411, Singapore
cambria@nus.edu.sg

Kenneth Kwok
NUS Temasek Laboratories
Cognitive Science Programme
117411, Singapore
kenkwok@nus.edu.sg

ABSTRACT

Topic modeling is a technique used for discovering the abstract 'topics' that occur in a collection of documents, which is useful for tasks such as text auto-categorization and opinion mining. In this paper, a commonsense knowledge based algorithm for document topic modeling is presented. In contrast to probabilistic models, the proposed approach does not involve training of any kind and does not depend on word co-occurrence or particular word distributions, making the algorithm effective on texts of any length and composition. 'Semantic atoms' are used to generate feature vectors for document concepts. These features are then clustered using group average agglomerative clustering, providing much improved performance over existing algorithms.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text Analysis*

General Terms

Algorithms

Keywords

AI, NLP, KR, Topic Modeling, Commonsense Knowledge

1. INTRODUCTION

Topics, defined as distributions over words [7], facilitate keyword-based text mining, document search, and meaning-based retrieval [4]. Probabilistic topic models, such as latent Dirichlet allocation [7], are used to facilitate document queries [39], document comprehension [27], and tag-based recommendations [22]. However, typical *bag-of-words* probabilistic models have several shortcomings.

WISDOM '13, August 11 2013, Chicago, USA.

Firstly, in such models words in a document are considered to be independent from each other, which is a very uncommon scenario in practice. For example, the multi-word expression *"score home run"*, taken as a unit, is clearly related to baseball, but word-by-word conveys totally different semantics. Similarly, in a bag-of-words model, the phrase *"getting fired"* [25] would not convey the meaning "fired from a job". Secondly, removing stopwords from documents often leads to the loss of key information, e.g., in the case of the concept *"get the better of"*, which would become *"get better"* with all stopwords removed, or the multi-word expression *"let go of the joy"*, which would be reduced to simply *"joy"*, reversing the meaning.

The alternative approach we put forth here draws on knowledge of *word meanings*, encoded in a commonsense knowledge database structured in the INTELNET formalism [32], to provide enhanced topic modeling capabilities. In contrast to probabilistic models, our approach does not involve training of any kind and does not depend on word co-occurrence or particular word distributions, making the algorithm effective on texts of any length and composition.

Commonsense knowledge, defined as the basic understanding of the world humans acquire through day-to-day life, includes information such as *"one becomes elated when one sees one's ideas become reality"* and *"working to be healthy is a positive goal"*. Statistical text models can identify email spam and find syntactic patterns in documents, but fail to understand poetry, simple stories, or even friendly e-mails. Commonsense knowledge is invaluable for nuanced understanding of data, with applications including textual affect sensing [24], handwritten text recognition [38], story telling [23], situation awareness in human-centric environments [20], casual conversation understanding [12], social media management [14], opinion mining [9], and more.

This paper proposes a method for using commonsense knowledge for topic modeling. Instead of the 'bag-of-words' model, we propose the *bag-of-concepts* [8], which considers each lexical item as an index to a set of 'semantic atoms' describing the concept referred to by that lexical item. A 'semantic atom' is a small piece of knowledge about a particular concept, perhaps that it tends to have a particular size or be

associated with other known concepts. Taken together, the set of semantic atoms for a concept provides an excellent picture of the practical meaning of that concept. Access to this knowledge permits the creation of 'smart' topic modeling algorithms that take semantics into account, offering improved performance over probabilistic models.

The paper is organized as follows: Section 2 describes related work in the field of topic modeling; Section 3 illustrates how the bags-of-concepts are extracted from natural language text; Section 4 discusses the proposed topic modeling algorithm; Section 5 presents evaluation results; finally, Section 6 concludes the paper and suggests directions for future work.

2. RELATED WORK

In topic modeling literature, a document is usually represented as a word vector $W = \{w_1, w_2, w_3 ....., w_n\}$ and the corpus as a set of documents $C = \{W_1, W_2, W_3 ....., W_n\}$, while the distribution of a topic z across C is usually referred as θ. Common approaches include mixture of unigrams, latent semantic indexing (LSI), and latent Dirichlet allocation (LDA).

2.1 Mixture of Unigrams

The mixture-of-unigrams model [30] is one of the earliest probabilistic approaches to topic modeling. It assumes that each document covers only a *single* topic (an assumption that does not often hold). The probability of a document in this model, constrained by the aim of generating N words independently, is defined as:

$$p\left(W\right) = \sum_{z=1}^{k} \left[\prod_{n=1}^{N} p\left(w_n|z\right) \right] p\left(z\right) \quad (1)$$

where $p\left(W\right)$ is the probability of the document, $p\left(w_n|z\right)$ is the probability of the word conditional on topic, and $p\left(z\right)$ is the probability of the topic. This model has the serious shortcoming of attempting to fit a single topic for the whole document.

2.2 Latent Semantic Indexing

A subsequent major development in topic modeling has been achieved through LSI, most commonly in the form of Hofmann's pLSI algorithm[19], given as:

$$p\left(W, w_n\right) = p\left(W\right) \sum_{z=1}^{k} p\left(w_n|z\right) p\left(z|W\right) \quad (2)$$

where $p\left(z|d\right)$ is the probability of the topic conditional on document. In pLSI, each document is modeled as a bag-of-words and each word is assumed to belong to a particular topic. The algorithm overcomes the shortcoming of the mixture-of-unigrams model by assuming that a document can cover multiple topics. However, it suffers from issues related to data overfitting.

2.3 Latent Dirichlet Allocation

LDA [7] is the state-of-the-art approach to topic modeling. It is a mixed-membership model, which posits that each document is a mixture of a small number of topics and that each word's creation is attributable to one of the document's topics. In particular, for a set of N topics:

$$p\left(w\right) = \int_{\theta} \left(\prod_{n=1}^{N} \sum_{z_n=1}^{k} p\left(w_n|z_n;\beta\right) p\left(z_n|\theta\right) \right) p(\theta;\alpha) d\theta \quad (3)$$

where α is the Dirichlet prior parameter per document topic distribution and β is the Dirichlet prior parameter per topic word distribution. Specifically, the LDA algorithm [5] operates as follows:

1. For each topic,
   (a) Draw a distribution over words $\overrightarrow{\beta_v} \sim Dir_v\left(\eta\right)$
2. For each document,
   (a) Draw a vector of topic proportion $\theta_d \sim Dir\left(\overrightarrow{\alpha}\right)$
   (b) For each word,
      i. Draw a topic assignment $Z_{d,n} \sim Mult\left(\overrightarrow{\theta_d}\right)$, $Z_{d,n} \in \{1, ... K\}$
      ii. Draw a word $W_{d,n} \sim Mult\left(\overrightarrow{\beta_{z_{d,n}}}\right)$, $W_{d,n} \in \{1, ... V\}$

To perform topic modeling for a particular task, posterior inference is perfomed using any one of the following methods: mean field variational [6], expectation propagation [28], collapsed Gibbs sampling [15] , distributed sampling [29], collapsed variational inference [37], online variational inference [18] and/or factorization-based inference [1].

Although LDA overcomes shortcomings such as overfitting and the presence of multiple topics within documents, it still uses the bag-of-words model and, hence, wrongly assumes that every word in the document is independent. For this reason, model performance is heavily dependent on topic diversity and corpus volume. LDA and other probabilistic models, in fact, fail to deal with short texts, unless they are trained extensively with all the documents covering the scope of the test document. The proposed commonsense-based model, instead, does not involve training of any kind and does not depend on word co-occurrence or particular word distributions, making the algorithm effective on texts of any length and composition.

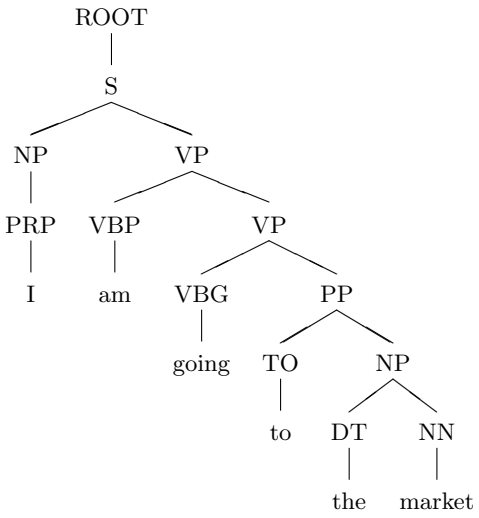
3. SEMANTIC PARSING

The first step to commonsense-based topic modeling is bag-of-concepts extracton. This is performed through a graph-based approach to commonsense concept extraction [34], which breaks sentences into chunks first and then extracts concepts by selecting the best match from a parse graph that maps all the multi-word expressions contained in the commonsense knowledge base.

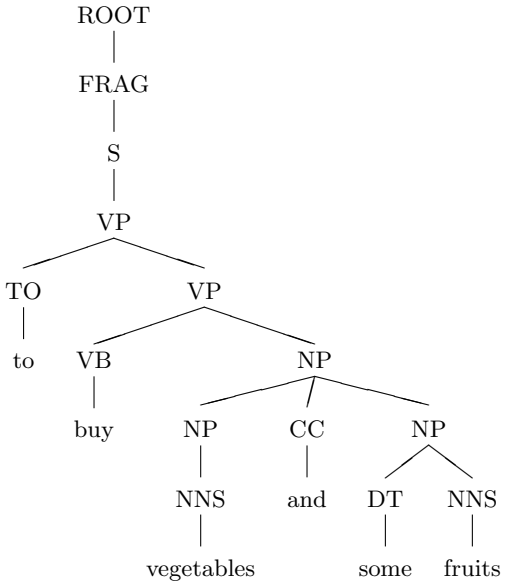
3.1 From Sentence to Verb and Noun Chunks

Each verb and its associated noun phrase are considered in turn, and one or more concepts is extracted from these. As an example, the clause "I went for a walk in the park", would contain the concepts *go walk* and *go park*.

The Stanford Chunker [26] is used to chunk the input text. A sentence like "I am going to the market to buy vegetables and some fruits" would be broken into "I am going to the market" and "to buy vegetables and some fruits". A general assumption during clause separation is that, if a piece of text contains a preposition or subordinating conjunction, the words preceding these function words are interpreted not as events but as objects. The next step of the algorithm then separates clauses into verb and noun chunks, as suggested by the following parse tree:



and



3.2 Obtaining the Full List of Concepts

Next, clauses are normalized in two stages. First, each *verb* chunk is normalized using the Lancaster stemming algorithm [33]. Second, each potential *noun* chunk associated with individual verb chunks is paired with the stemmed verb in order to detect multi-word expressions of the form 'verb plus object'.

```
Data: NounPhrase
Result: Valid object concepts
Split the NounPhrase into bigrams ;
Initialize concepts to Null ;
for each NounPhrase do
    while For every bigram in the NounPhrase do
        POS Tag the Bigram ;
        if adj noun then
            add to Concepts: noun, adj+noun

        else if noun noun then
            add to Concepts: noun+noun

        else if stopword noun then
            add to Concepts: noun

        else if adj stopword then
            continue

        else if stopword adj then
            continue

        else
            Add to Concepts : entire bigram
        end
        repeat until no more bigrams left;
    end
end
```

Algorithm 1: POS-based bigram algorithm

Objects alone, however, can also represent a commonsense concept. To detect such expressions, a POS-based bigram algorithm checks noun phrases for stopwords and adjectives. In particular, noun phrases are first split into bigrams and then processed through POS patterns, as shown in Algorithm 1. POS pairs are taken into account as follows:

1. ADJECTIVE NOUN : The adj+noun combination and noun as a stand-alone concept are added to the objects list.
2. ADJECTIVE STOPWORD : The entire bigram is discarded.
3. NOUN ADJECTIVE : As trailing adjectives do not tend to carry sufficient information, the adjective is discarded and only the noun is added as a valid concept.
4. NOUN NOUN : When two nouns occur in sequence, they are considered to be part of a single concept. Examples include *butter scotch*, *ice cream*, *cream biscuit*, and so on.
5. NOUN STOPWORD : The stopword is discarded, and only the noun is considered valid.
6. STOPWORD ADJECTIVE: The entire bigram is discarded.
7. STOPWORD NOUN : In bigrams matching this pattern, the stopword is discarded and the noun alone qualifies as a valid concept.

Data: Natural language sentence
Result: List of concepts
Find the number of verbs in the sentence;
for *every clause* **do**
  extract VerbPhrases and NounPhrases;
  stem VERB ;
  for *every NounPhrase with the associated verb* **do**
    find possible forms of *objects* ;
    link all *objects* to stemmed verb to get *events*;
  end
  repeat until no more clauses are left;
end

Algorithm 2: Event concept extraction algorithm

The POS-based bigram algorithm extracts concepts such as *market*, *some fruits*, *fruits*, and *vegetables*. In order to capture event concepts, matches between the object concepts and the normalized verb chunks are searched. This is done by exploiting a parse graph that maps all the multi-word expressions contained in the knowledge bases (Fig. 1). Such an unweighted directed graph helps to quickly detect multi-word concepts, without performing an exhaustive search throughout all the possible word combinations that can form a commonsense concept.

Single-word concepts, e.g., *house*, that already appear in the clause as a multi-word concept, e.g., *beautiful house*, in fact, are pleonastic (providing redundant information) and are discarded. In this way, algorithm 2 is able to extract event concepts such as *go market*, *buy some fruits*, *buy fruits*, and *buy vegetables*, representing SBoCs to be fed to a commonsense reasoning algorithm for further processing.

4. TOPIC MODELING ALGORITHM

Once natural language text is deconstructed into bags-of-concepts by means of the graph-based approach to commonsense concept extraction, the topic modeling algorithm determines the final topic set, according to the semantic features associated with the input text.

In the INTELNET architecture [32], knowledge base concepts are defined by the ways in which they *interact* with one another, and semantic features are derived from interconnections as described below. The nuanced structure of the knowledge representation enables us to select just that information most likely to help achieve specific tasks. The rich semantic tapestry generated by the knowledge representation allows us to interpret documents as *collections of inter-connected concepts* rather than independent bags-of-words. The system presently only considers concepts that are in the knowledge base, but as coverage is quite extensive (currently around 9M pieces of information). Knowledge base concepts are collected from sources including ConceptNet [35], DBPedia [2], NELL [10], and WordNet [13].

The presence of a very wide diversity of concepts in the knowledge base makes the model effective for any generic dataset (newspaper articles, reviews, and so on). Documents do not need to be grammatically well-formed, and the algorithm's simultaneous consideration of the semantics of multiple concepts at the same time makes it resistant to

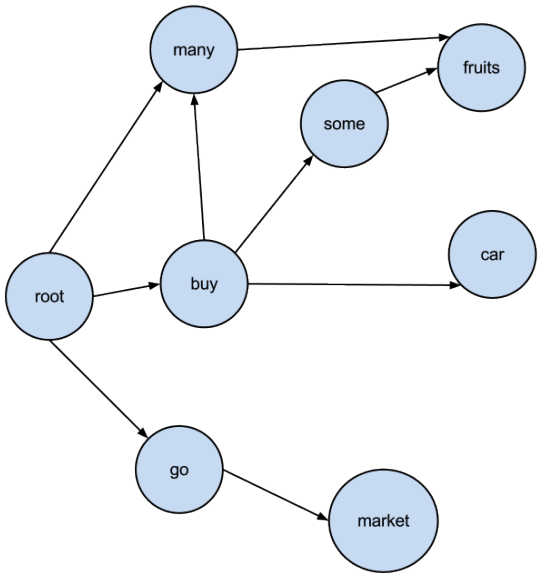


Figure 1: Sample parse graph for multi-word expressions in the knowledge base

spelling errors. Even if one concept is misspelled, it is likely that others from the same topic will be present, thus filling the gap. In domains where documents may be frequently expected to employ concepts not present in general commonsense knowledge bases, a key benefit of INTELNET is its ability to absorb new and noisy knowledge without affecting other knowledge. In these domains, specialized knowledge extracts may be added to the database in order to further enhance performance without damaging general capabilities. These capabilities make feature extraction an ideal base for clustering algorithms.

4.1 The Bag-of-Concepts Model

In our model, every document is represented as a *bag-of-concepts*, where concepts may be single lexical items or multi-word expressions. Phrases like *"get good grade"* are considered to be single concepts, maintaining the semantic interrelatedness of constituent lexical items, unlike the bag-of-words model where every single word is considered to be independent. Documents are defined as the union of the set of commonsense knowledge items retrieved for each individual document concept.

The bag-of-concepts model is different from other language modeling techniques such as *N-grams* wherein the probability of a sequence of words is calculated using the conditional probability of previous words. With bag-of-concepts, the unique semantics attached to particular combinations of words are retained and used to enhance algorithm performance. Specific word sequences evoke unique commonsense concepts together with the particularized semantics attached to those sequences.

The commonsense reasoning framework acts as a hidden layer of knowledge. Concepts in the knowledge base are selected by document contents and are semantically connected to one another, providing a rich data source for clustering.

4.2 Commonsense-Based Feature Extraction

Topics exhibit semantic coherence in that they tend to generate lexical items and phrases with related semantics. Most words related to the same topic tend to share some semantic characteristics. Our commonsense-based approach is similar to the process undertaken by humans when finding similar items - we look at what the *meanings* of the items have in common.

Thus, under our model, *topics* are not discovered merely based on document co-occurrence, but rather by considering the definitive semantic character of constituent concepts.

In INTELNET knowledge bases, concepts inter-define one another, with directed edges indicating semantic dependencies between concepts. In the present algorithm, the features for any particular concept C are defined as the set of concepts reachable via outbound edges from C. Put differently, for each input concept we retrieve those other concepts which, collectively, generate the core semantics of the input concept.

4.3 Clustering and Topic Detection

Our algorithm uses clustering to generate topics from semantic features. Based on experiments with various clustering algorithms, e.g., k-means [17] and expectation-maximization (EM) clustering [11], we determined that group average agglomerative clustering (GAAC) provides the highest accuracy.

GAAC partitions data into trees [3] containing *child* and *sibling* clusters. It generates dendrograms specifying nested groupings of data at various levels [21]. During clustering, documents are represented as vectors of commonsense concepts. For each concept, the corresponding features are extracted from the knowledge base. The proximity matrix is constructed with concepts as rows and features as columns. If a feature is an outbound link of a concept, the corresponding entry in the matrix is 1, otherwise it is 0. Cosine distance is used as the distance metric.

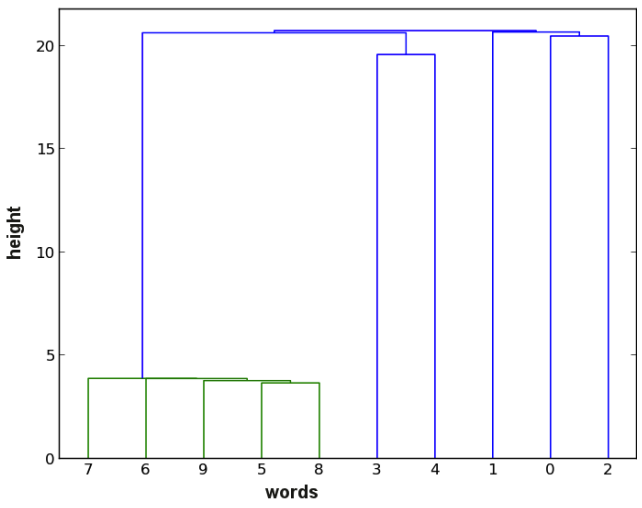


Figure 2: A sample dendrogram resulting from hierarchical clustering.

"horse"	"stationery"	"food"	"party"
horse	paper	apple	dance
eye	paint	fish	protest
farm	plate	bread	music
	card	cake	party
	metal		door
			sound
			weather
			wind

Table 1: Example: Feature-Based Clustering

Agglomerative algorithms are bottom-up in nature. GAAC consists of the following steps:

1. Compute proximity matrix. Each data item is an initial cluster.
2. From the proximity matrix, form pair of clusters by merging. Update proximity matrix to reflect merges.
3. Repeat until all clusters are merged.

A sample dendrogram is shown in Figure 2. The dendogram is pruned at a height depending on the number of desired clusters. The *group average* between the clusters is given by the average similarity distance between the groups. Distances between two clusters and similarity measures are given by the equations below:

$$X_{sum} = \sum_{d_m \in \omega_i \upsilon \omega_i} \sum_{d_n \in \omega_i \upsilon \omega_j d_n \neq d_m} \overrightarrow{d_n}.\overrightarrow{d_m} \tag{4}$$

$$sim\,(\omega_i, \omega_j) = \frac{1}{(N_i + N_j)\,(N_i + N_j - 1)} X_{sum} \tag{5}$$

where $\overrightarrow{d}$ is the vector of document of length d. Vector entries are boolean, 1 if the feature is present, 0 otherwise. N_i, N_j is the number of features in ω_i and ω_j respectively, which denote clusters.

The main drawback of the hierarchical clustering algorithm is the running complexity [3], which averages $\theta(N^2 log\ N)$.

We choose average link clustering because our clustering is connectivity-based. The concept proximity matrix consists of features from the knowledge base and 'good' connections occur when two concepts share multiple features.

After clustering, the number of clusters are determined and the dendrogram is pruned accordingly. The output of this process is the set of topics present in the document.

Table 1 provides an example of the results of feature-based clustering.

Topic	Latent Dirichlet Allocation			Commonsense with GAAC		
	Precision	Recall	F-measure	Precision	Recall	F-measure
1	0.67	0.2	0.31	0.875	0.7	0.78
2	0.58	0.67	0.62	0.86	0.67	0.75
3	0.3	0.2	0.24	1.0	0.98	0.99

Table 2: Results for *News* article

Topic	Latent Dirichlet Allocation			Commonsense with GAAC		
	Precision	Recall	F-measure	Precision	Recall	F-measure
1	0.25	0.2	0.22	1.0	0.625	0.77
2	0.12	0.4	0.18	0.27	0.8	0.4
3	0.2	0.31	0.24	0.58	0.85	0.69

Table 3: Results for *Religion* article

5. EXPERIMENTAL RESULTS

5.1 Evaluation of Cluster Quality

Evaluating topic modeling algorithms is not straight-forward. Evaluation measures should seek to compare apples to apples, yet existing algorithms for topic modeling are all either statistical or probabilistic in nature. Therefore, we rely on standard measures (*precision*, *recall* and *F-measure*) [36] to evaluate our model.

These measures are defined as follows:

$$\text{recall} = \frac{TP}{TP + FN} \qquad \text{precision} = \frac{TP}{TP + FP}$$

$$\text{TP rate} = \text{recall}$$

$$\text{FP rate} = \frac{FP}{FP + TN}$$

$$\text{F-measure} = 2 * \frac{precision * recall}{precision + recall}$$

For the probabilistic models, precision and recall are calculated based on the topics in which words occur with the highest probability.

5.2 Experimental Setup

Our testing dataset was derived from the Brown Corpus. Two 300-word test sets, each with three separate topics, were extracted from the Brown categories *Religion* and *News*. Religion topics included 'Nation/Country', 'Christianity', and 'Economy/Politics', and news topics included 'body/injury', 'game', and 'places'. The number of clusters was set to 7 for evaluation purposes. Given that LDA provided the best performance of all pre-existing topic modeling algorithms, we used it as the baseline for comparison. Results are shown in Table 2.

5.3 Discussion

It is difficult to compare models that rely on training, such as LDA, and those that do not, including the method proposed here. Firstly, the large amount of knowledge available to our method could be seen as an unfair advantage.

Secondly, it is difficult to determine a training set that would be comparable to the knowledge available in our commonsense database. Beyond this, short documents present issues for LDA as they do not provide sufficient text to make co-occurrence a useful metric.

For any statistical text mining algorithm, test sets should be independent of the training data but follow the same probability distribution as that data. Our algorithm does not consider probability distributions, and thus is independent of any such requirement.

In order to attempt to overcome these issues, training of the LDA model was attempted with the entire Brown corpus, as training with the entire commonsense knowledge base would have resulted in an overwhelmingly large number of potential topics. Trained LDA did not perform well, however, because relevant words in the test document were not likely to appear in the topic list given that their relative level of co-occurrence for each topic in the corpus was quite low. For comparison, an HMM-LDA model [16] (trigram) was also tested, but this model suffered from the same problems (although at a larger scale due to the inclusion of bigrams and trigrams). Thus, untrained LDA was used for comparison in Tables 2 and 3.

6. CONCLUSION

We have presented a commonsense-based topic modeling algorithm using INTELNET 'semantic atoms' and clustering to determine the topics present in particular documents. We also provide evidence of the algorithm's improved performance over the state-of-the-art algorithm.

One key future extension of this work is to use *language modeling* in place of the current semantic parsing algorithm. In particular, we plan to use topic knowledge in conjunction with the COGPARSE [31] semantic parsing engine to identify semantic and syntactic patterns based on topic information, driving a new approach to language modeling.

This would allow the narrowing of searches to specific paragraphs, taking word order into account and allowing highly targeted searches at sub-document scopes.

7. REFERENCES

[1] S. Arora, R. Ge, and A. Moitra. Learning topic models–going beyond svd. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 1–10. IEEE, 2012.

[2] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[3] P. Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer, 2006.

[4] D. M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[5] D. M. Blei and J. D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[6] D. M. Blei and P. J. Moreno. Topic segmentation with an aspect hidden markov model. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 343–348. ACM, 2001.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3:993–1022, 2003.

[8] E. Cambria and A. Hussain. *Sentic Computing: Techniques, Tools, and Applications*. Springer, Dordrecht, Netherlands, 2012.

[9] E. Cambria, B. Schuller, Y. Xia, and C. Havasi. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, 28(2):15–21, 2013.

[10] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka Jr, and T. M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence (AAAI 2010)*, volume 2, pages 3–3, 2010.

[11] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal statistical Society, Series B*, 39(1):1–38, 1977.

[12] N. Eagle, P. Singh, and A. Pentland. Common sense conversations: understanding casual conversation using a common sense database. In *Proceedings of the Artificial Intelligence, Information Access, and Mobile Computing Workshop (IJCAI 2003)*, 2003.

[13] C. Fellbaum. *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, 1998.

[14] M. Grassi, E. Cambria, A. Hussain, and F. Piazza. Sentic web: A new paradigm for managing social media affective information. *Cognitive Computation*, 3(3):480–489, 2011.

[15] T. Griffiths and M. Steyvers. A probabilistic approach to semantic representation. In *Proceedings of the 24th annual conference of the cognitive science society*, pages 381–386. Citeseer, 2002.

[16] T. Griffiths, M. Steyvers, D. M. Blei, and J. Tenenbaum. *Integrating topics and syntax*, volume 17 of *Advances in Neural Information Processing Systems*, pages 537–544. MIT Press, Cambridge, MA, 2005.

[17] J. Hartigan and M. Wong. Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society*, 28(1):100–108, 1979.

[18] M. Hoffman, D. M. Blei, and F. Bach. Online learning for latent dirichlet allocation. *Advances in Neural Information Processing Systems*, 23:856–864, 2010.

[19] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[20] N. Howard and E. Cambria. Intention awareness: Improving upon situation awareness in human-centric environments. *Human-centric Computing and Information Sciences*, 3(9), 2013.

[21] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[22] R. Krestel, P. Fankhauser, and W. Nejdl. Latent dirichlet allocation for tag recommendation. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 61–68, New York, NY, USA, 2009. ACM.

[23] H. Lieberman, H. Liu, P. Singh, and B. Barry. Beating common sense into interactive applications. *AI Magazine*, 25(4):63, 2004.

[24] H. Liu, H. Lieberman, and T. Selker. A model of textual affect sensing using real-world knowledge. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 125–132. ACM, 2003.

[25] H. Liu and P. Singh. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004.

[26] C. Manning. Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In A. Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*, volume 6608 of *Lecture Notes in Computer Science*, pages 171–189. Springer, 2011.

[27] G. Maskeri, S. Sarkar, and K. Heafield. Mining business topics in source code using latent dirichlet allocation. In *Proceedings of the 1st India software engineering conference*, pages 113–120. ACM, 2008.

[28] T. Minka and J. Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 352–359. Morgan Kaufmann Publishers Inc., 2002.

[29] D. Newman, A. Asuncion, P. Smyth, and M. Welling. Distributed inference for latent dirichlet allocation. *Advances in neural information processing systems*, 20(1081-1088):17–24, 2007.

[30] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2-3):103–134, 2000.

[31] D. Olsher. COGPARSE: Brain-inspired knowledge-driven full semantics parsing: Radical construction grammar, categories, knowledge-based parsing & representation. In *Advances in Brain Inspired Cognitive Systems*, volume 7366 of *LNCS*. Springer, 2012.

[32] D. Olsher. COGVIEW & INTELNET: Nuanced energy-based knowledge representation and integrated cognitive-conceptual framework for realistic culture,

values, and concept-affected systems simulation. In *Proceedings, 2013 IEEE Symposium Series on Computational Intelligence*, 2013.

[33] C. Paice. Another stemmer. *SIGIR Forum*, 24(3):56–61, 1990.

[34] D. Rajagopal, E. Cambria, D. Olsher, and K. Kwok. A graph-based approach to commonsense concept extraction and semantic similarity detection. In *WWW*, pages 565–570, Rio De Janeiro, 2013.

[35] R. Speer and C. Havasi. ConceptNet 5: A large semantic network for relational knowledge. In E. Hovy, M. Johnson, and G. Hirst, editors, *Theory and Applications of Natural Language Processing*, chapter 6. Springer, 2012.

[36] M. Steinbach, G. Karypis, V. Kumar, et al. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, 2000.

[37] Y. W. Teh, D. Newman, and M. Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, volume 19, 2007.

[38] Q. Wang, E. Cambria, C. Liu, and A. Hussain. Common sense knowledge for handwritten chinese recognition. *Cognitive Computation*, 5(2):234–242, 2013.

[39] X. Wei and W. B. Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185. ACM, 2006.

Online Debate Summarization using Topic Directed Sentiment Analysis

Sarvesh Ranade, Jayant Gupta, Vasudeva Varma, Radhika Mamidi
International Institute of Information Technology, Hyderabad
{sarvesh.ranade | jayant.gupta}@research.iiit.ac.in
{vv | radhika.mamidi}@iiit.ac.in

ABSTRACT

Social networking sites provide users a virtual community interaction platform to share their thoughts, life experiences and opinions. Online debate forum is one such platform where people can take a stance and argue in support or opposition of debate topics. An important feature of such forums is that, they are dynamic and increase rapidly. In such situations, effective opinion summarization approaches are needed so that readers need not go through the entire debate. This paper aims to summarize online debates by extracting highly topic relevant and sentiment rich sentences. The proposed approach takes into account topic relevant, document relevant and sentiment based features to capture topic opinionated sentences. ROUGE scores are used to evaluate our system. Our system significantly outperforms several baseline systems and show 5.2% (ROUGE-1), 7.3% (ROUGE-2) and 5.5% (ROUGE-L) improvement over the state-of-the-art opinion summarization system. The results verify that topic directed sentiment features are most important to generate effective debate summaries.

1. INTRODUCTION

With the exponential growth in the use of World Wide Web, online users express themselves on continuously emerging social networking sites. These sites provide users a wide variety of choices. Micro-blogging sites like twitter allows them to express their opinion (140 characters) on trending topics. Users can express their views and share their experiences on popular web blogs like wordpress, blogspot, etc. Facebook allows community oriented interaction, restricted within one's friend circle. E-commerce users can provide product specific reviews on online shopping sites.

Amongst social networking platforms, online debate sites ('convinceme.net', '4forums.net', 'onlinedebate.net') have become popular in recent times. These online debate forums provide users an option to express their opinion about their favorite debate topics [28]. From the research point of view, they provide a rich collection of differing opinions on various topics. In ideological two-sided debates, users support their stance by cleverly stating arguments supporting their stance or opposing other stance [27]. Conversation sentences between two users are very common in this multi-party conversation. People rebut to another user's post and express their viewpoint on other's opinion [2]. Because of dynamic nature of debates and large number of posts (194 per debate) they contain, it is essential to generate effective summaries for them so that readers need not go through the entire debate.

To help this cause, we need to summarize online debates such that, after reading the summaries, user gets a good idea about the information debate presents and the opinions users express. This paper aims to summarize online debates collected from a popular debate site called 'convinceme.net' with the intention of capturing good topic information as well as opinion rich sentences from the debate.

Compared to generic summarization, opinion summarization is a relatively novel area. Unlike traditional methods, two factors, the sentiment degree and the correlated events, play a major role in opinion summarization. Previous methods [29, 15] have effectively used these factors over news, blogs and conversation domain. However, online debates is a domain which is yet to be explored. An important aspect of online debates is that most of its sentences are sentiment rich and topic relevant. Thus, topic directed sentiment analysis is an important feature to create effective summaries. We have successfully used this feature and results validate the effectiveness of our approach.

In our method, we analyzed factors governing important sentences in debate summaries. We observed 3 important factors: informative sentences, sentiment rich sentences and sentences which describes topic related entities. Thus, topic related and sentiment carrying features are used in the proposed approach. We have also used positional features as they have been effective in generic summarization approaches. Document relevant features such as tf-idf scores are used to capture content relevant sentences.

Our system, $DEBSumm$ generates extractive summaries using the aforementioned features. ROUGE [18] scores have been used to evaluate the system summaries. Final results show that sentiment words that are relevant to topic are the most important feature to create effective summaries. The results also show that our system achieve better results than previous state-of-the-art and several baseline systems.

The rest of the paper is organized as follows: Section 2 describes related work; Section 3 gives a detailed description of our approach. Section 4 describes experimental setup for

experiments; Section 5 discuss results of different experiments performed; Section 6 concludes the paper with final comments and future work.

2. RELATED WORK

An extensive work has been done in the field of extractive text summarization. Earliest work by Luhn [19] used frequency based features to score sentences. Later work [9, 16] added features such as topic signature, cue words, data annotation, etc. These features were used to score sentences and top sentences were selected for summary. In MEAD [25], clusters were created and sentences were scored using sentence and inter sentence features. Redundancy removal has been a big issue in summarization for which a benchmark approach was proposed by Carbonnell et al. [7]. This approach used MMR to create a balance between information novelty and importance to create non redundant summaries.Graph based approaches [26, 20] represent text as a graph where a text entity (sentence, phrases) represent vertices connected by similarity based measures. Salton et al. [26] use cosine similarity to connect vertices, then a greedy graph traversal technique is applied in chronological order to form summary.

In context of opinion summarization, identification of opinion containing sentences is important. Sentence relevance is further decided by their sentiment scores, topic relevance and other lexical and positional features. Earlier works mainly focused on reviews [24, 13, 22] which used lexical features (unigram, bigram and trigram), part-of-speech tags and dependency relations.

Ku et al. [15] performed opinion summarization in news and blog domain. They propose opinion extraction at word, sentence and document level. For each new word, distribution of its characters (Chinese) as positive and negative polarity in the seed vocabulary (created manually) is used to determine sentiment of the word. These scores are compounded to compute sentence scores and then document scores. Presence of negation operators decided the sentiment tendency at sentence level which further propagated to document level.

Wang et al. [29] performed opinion summarization on conversations. They used linear combination of features from different aspects including topic relevance, subjectivity and sentence importance to score sentences. They also proposed a graph based method, which incorporates topic and sentiment information, as well as additional information about sentence to sentence relations extracted based on dialogue structures.

Mining of opinion goes hand in hand with analyzing sentiments. In this perspective, a detailed study of past work, present trends and futures needs has been done by Cambria et al. [5]. Significant work has been done in social media on target directed sentiment analysis [1, 23, 21]. Agarwal et al. [1] used syntactic features as target dependent features to differentiate sentiment words' effect on different targets in a tweet. O'Hare et al. [23] employed word-based approach to calculate sentiments directed towards companies and their stocks from financial blogs.

Cambria et al. [6] applied semantic multi-dimensional scaling on a knowledge base of affective common-sense knowledge for text classification, emotion recognition, and patient opinion mining. Mukherjee et al. [21] applied clustering to extract feature specific opinions and calculated overall feature sentiment using subjectivity lexicon.

Opinion summarization in the specific domain of online debates is a novel field. This domain differs from chatting and conversation because it is more formal and focuses on specific topics. It may be possible that the argument contains various different factual knowledge but they are usually related to one or the other topic. Similarly, it is different from news and blogs because it is comparatively more rich in sentiment. Therefore, opinion mining in debates is an interesting and challenging task.

3. APPROACH

Extractive summaries are generated by ranking the Dialogue Acts (DAs)[1] from the original documents. We calculate their importance according to linear combination of scores using several features. Equation 1 is used to assign score to each DA s.

$$score(s) = \lambda_{topicRel} topicRel(s, topics) + \lambda_{docRel} docRel(s, D) + \lambda_{sentiRel} sentiRel(s) + \lambda_{conRel} conRel(s, D) \quad (1)$$

Most highly ranked DAs are chosen until summary length constraint is satisfied. Table 1 lists the set of features used in this equation. We describe each of these features in the subsequent subsections.

Feature Category	Feature Names
Topic Relevance	Topic Directed Sentiment Score Topic Co-occurrence
Document Relevance	tf-idf Sentiment Score
Sentiment Relevance	Number of Sentiment Words Sentiment Strength
Context Relevance	Sentence position Sentence length

Table 1: Argument Structure Examples

3.1 Topic Relevant Features

Debate posts present users' opinion towards debate topics. Thus, sentences which provide information or express opinion about debate topics are most important in the context of debate summarization. We use topic directed sentiment scores and topic co-occurrence measure to capture topic relevance of the DAs.

3.1.1 Topic Directed Sentiment Score

Topic related sentiment carrying DAs are very important in the context of online debates. They represent the sentiments directed by DA toward debate topics and thus, a key feature in the task of debate summarization. In the proposed approach, the sentiment score directed towards debate topics is calculated using dependency parse of the DAs and sentiment lexicon *SentiWordNet [4].*

Pronoun referencing is resolved using Stanford co-reference resolution system [17]. Then using Stanford dependency parse [8], DAs are represented in tree format where each node represents a DA word storing its sentiment score and the edges represent dependency relations. Each DA word is looked in SentiWordNet and the sentiment score calculated using Algorithm 1 is stored in the word's tree node.

[1]Dialogue Act is smallest unit of debate.

Algorithm 1 Word Sentiment Score

1: $S \leftarrow Senses\ of\ word\ W$
2: $wordScore \leftarrow 0$
3: **for all** $s \in S$ **do**
4: $\quad s_{score} = s_{posScore} - s_{negscore}$
5: $\quad wordScore = wordScore + s_{score}$
6: **end for**
7: $wordScore = \frac{wordScore}{|S|}$

SentiWordNet is a lexical corpus used for opinion mining. It stores positive and negative sentiment scores for every sense of the word present in WordNet [10]. For words missing from SentiWordNet, average of sentiment scores of its synset member words is stored in the word's tree node, otherwise zero sentiment score is stored. If words are modified by negation words like {'never','not','nonetheless', etc.}, their sentiment scores are negated.

In noun phrases 'great warrior', 'cruel person', etc. first word being the adjective of the latter, influences its sentiment score. Thus, based on the semantic significance of the dependency relation each edge holds, sentiment score of parent nodes are updated with that of child nodes using Algorithm 2. In DAs like "Batman killed a bad guy.", sentiment score of word 'Batman' is affected by action 'kill'. Thus, for verb-predicate relations like {'nsubj','dobj','cobj','iobj', etc.}, predicate sentiment scores are updated with that of verb scores using Algorithm 2.

Algorithm 2 Update Word Sentiment Score

1: $node \leftarrow Word's\ Tree\ Node$
2: $childs \leftarrow Word's\ child\ nodes$
3: **for all** $c \in childs$ **do**
4: $\quad updateScore(c)$
5: $\quad node_{score} \leftarrow sign(node_{score}) * (|node_{score}| + (c_{score}))$
6: **end for**

Tree structure and recursive nature of Algorithm 2 ensures that sentiment scores of child nodes are updated before updating their parents' sentiment scores. Table 2 lists the semantically significant dependency relations used to update parent node scores.

Modification Type	Dependency Relations
Noun Modifying	nn, amod, appos, abbrev, infmod, poss, rcmod, rel, prep
Verb Modifying	advmod, acomp, advcl, ccomp, prt, purpcl, xcomp, parataxis, prep

Table 2: List of Dependency Relations

Extended Targets (ET): Extended targets are the entities closely related to debate topics. For example, 'Joker', 'Clarke Kent' are related to 'Batman' and 'Darth Vader', 'Yoda' to 'Star Wars'. To extract the extended targets, we capture named entities (NE) from Wikipedia page of the debate topic using Stanford Named Entity Recognizer [11] and sort them based on their frequency. Out of *top-k* ($k = 20$) NEs, some can belong to both of the debate topics. For example, 'DC Comics' is common between 'Superman' and 'Batman'. We remove these NEs from individual lists and the remaining NEs are treated as extended targets (*extendedTargets*) of the debate topics.

Now that we have a list of extended targets for debate topics and a sentiment score for each DA word, topic directed sentiment scores are calculated for each debate topic using Equation 2.

$$Topic\ Score_{DA} = \sum_{\substack{w \in DA \\ w \in ET(Topic)}} (Score(w)) \quad (2)$$

We refer to these scores as *AScore* and *BScore* representing scores directed towards topics A and B in debate between these two topics respectively.

Absolute value of both topic directed sentiment scores are added reprensenting DA's topic directed sentiment score. These scores are normalized with the sum of topic directed sentiment score of all the DAs.

3.1.2 Topic Co-occurrence Measure

Topic co-occurrence measure captures DAs containing high sentiment words which highly co-occur with debate topic. Extended targets previously described represent debate topic entities. Topic co-occurrence measure is computed using HAL from the Equation 3, capturing co-occurrence measure of DA words and their sentiment strengths. Sentiment score is calculated using Algorithm 1.

$$Co-occur_{DA} = \sum_{w \in DA} \left(\sum_{t \in ET} (HAL(w|t)) * sentiScore(w) \right) \quad (3)$$

Topic-occurrence measure is normalized with the sum of co-occurrence scores of all the DAs. We sum up topic directed sentiment scores and topic co-occurrence measure giving us the topic relevance feature score for DAs.

3.2 Document Relevance Features

Tf-idf and sentiment score of the words are used to compute document relevance of the DAs using Equation 4.

$$tf - idf_{DA} = \sum_{w\ \in\ DA} (tf - idf(w)\ *\ sentiScore(w)) \quad (4)$$

Tf-idf score reflects how important a word is to a document in a collection or corpus. Sentiment score carrying words' sentiment strength reflects subjective importance of the word in the context of opinion DAs. Thus, this feature captures the DAs containing highly frequent sentiment rich words. Document relevance score of the whole debate DAs is used to normalize individual scores.

3.3 Sentiment Relevance Features

This dimension captures the presence of sentiment carrying words and their strength in the DAs.

1. *sentiCount* is the count of sentiment carrying words in the DAs. *sentiCount* is normalized with total number of sentiment words present in the debate.

2. Sentiment score of each DA word is calculated using Algorithm 1 and Equation 5 is used to compute DAs' sentiment strength. Sentiment score for each DA is normalized with overall debate's sentiment score.

$$sentiScore_{DA} = \sum_{w \in DA} sentiScore(w) \quad (5)$$

Sentiment score and number of sentiment words in DAs are added which represents the sentiment relevance feature score of DAs.

3.4 Document Context Features

3.4.1 Sentence Position

Sentence position plays important role in predicting the presence of DAs in summary. In debates, initial and ending DAs of the debate posts are more important than the middle ones. So, we have used Equation 6 to compute sentences' position based score which gives higher values for initial and ending sentence than the middle ones. This score is normalized by dividing it with number of DAs in debate posts[2].

$$posScore_{DA} = \frac{|\ \frac{N}{2} - DA_{position}\ |}{N}\ , N = Total\ DAs\ in\ Post \quad (6)$$

3.4.2 Sentence Length

As the longer sentences tend to contain more information, we have used sentence length as document context feature. It also avoids short sentences (smaller than 5 words) which are less likely to contribute to summary because of incompleteness or less information. Sentence length is the number of words in the DAs. We have normalized the sentence length with the number of words in the whole debate.

We sum sentence position and sentence length scores to compute document context feature score of DAs.

Note that all the values have been normalized over all DAs in the debate so that the different feature scores are comparable.

4. EXPERIMENTAL SETUP

In this study, we extracted 10 online debate discussions from *www.convinceme.net*. These discussions are freely available on aforementioned site and Table 3 shows the statistics of the dataset used. Each of these discussions focus upon different topics allowing us to produce results over various domains.

Number of users	Number of posts	Number of DA
1168	1945	23681

Table 3: Statistics of the dataset

For evaluation, extractive gold set summaries were created by 2 language editors. They were asked to create 500, 1000, 1500, 2000 word summaries. Inter-editor agreement was calculated to be 71.7%[3]. The editors were asked to select the sentences on the following order:

1. Sentiment rich which contains highly topic-relevant information.
2. Sentiment rich with relevant information (low noise).
3. Less subjective content but rich in information.
4. Highly subjective sentence with no relevant information and factual statements should be selected with care. The reason being that they add noise without taking any particular stand.

All the evaluation scores are computed using ROUGE [18] which stands for Recall Oriented Understudy of Gisting Evaluation. It has been widely used by DUC to evaluate system summaries. ROUGE measures summary quality by counting overlapping units such as the n-gram, word-sequences and word-pairs between system summaries and human summaries. Three automatic evaluation methods ROUGE-1, ROUGE-2 and ROUGE-L were chosen to calculate scores. They compute unigram recall, bigram recall and longest common subsequence respectively.

We have conducted the following experiments :

1. Comparison of $DEBSumm$ summaries with proven baseline and state-of-the-art summarization systems explained in Section 5.
2. Effect of variable summary size on $DEBSumm$ and state-of-the-art systems.

5. RESULTS AND DISCUSSION

Grid search was used to compute best parameter values for Equation 1. Following values gave the best results as indicated by ROUGE results: $\lambda_{topicRel} = 0.3, \lambda_{docRel} = 0.1, \lambda_{sentiRel} = 0.5, \lambda_{conRel} = 0.1$.[4]

Scores show that better summaries are obtained when sentiment rich sentences are selected. Furthermore, sentiments which are directed towards the topic words are also given higher weightage. Other measures like sentence position and length give a better fine tuning to summaries as they help to differentiate between similar sentences. Low weightage to document relevance score is understandable because it is a redundant feature to identify sentiment rich document words.

We compared our system ($DEBSumm$) to the following systems:

1. **Max-length [12]:** Longest sentences were selected from all the users. In case, summary is short of length second-longest sentences are selected. This step is iterated until summary reaches required length. This is a proven strong baseline for conversation summarization.
2. **Lead [30]:** Top sentences from each user were selected where each sentence has to be greater than 4 words. In case, summary is short of length, next sentence is selected. This step is iterated until summary reaches required length.
3. **pHAL [14]:** Sentence (S) score was calculated by combining the pHAL scores of each of sentence words. pHAL score of each word is calculated as follows,

$$pHAL(w) = \sum_{w' \epsilon ET} \frac{HAL(w'|w)}{n(w) * K}$$

$$Score(S) = \sum_{w_i \epsilon S} (P(w_i) \times pHAL(w_i))$$

[2]Post represents a user argument and consists of multiple DAs

[3]Number of common sentences were averaged over the complete set of debates.

[4]All the further experiments were conducted using these values.

For summary creation, top scored sentences were selected from sorted list of sentences.

4. **tf-Idf [3]:** Sentences were scored by combining the tf-idf measures of their words[5]. For summary creation, top most sentences were selected from sorted list of sentences.

5. **OpinionSumm [29]:**[6] This is a sentence scoring approach where sentence are scored based on their document similarity, topic relevance, sentiment and length. We have used the same parameter values experimentally calculated in their work. This is a state-of-the-art opinion summarization system.

In the field of generic summarization, system 2 and 4 are proven strong baseline and system 3 is a state-of-the-art system.

Table 4: ROUGE Scores (Average F-measure) of System Summaries (1000 words)

System	ROUGE-1	ROUGE-2	ROUGE-L
Max-Length	0.49892	0.18453	0.48343
Lead	0.49068	0.14759	0.47839
pHAL	0.48985	0.16468	0.46955
tf-idf	0.49922	0.17585	0.48035
OpinionSumm	0.51631	0.20364	0.49849
DEBSumm	**0.56833**	**0.27044**	**0.55326**

Table 4 shows ROUGE scores (Average F-measure) of different systems. The summary size is taken to be 1000 words. Note that, each of the systems 1, 2, 3 and 4, is one of the lower weighted components of the function used to compute our (*DEBSumm*) scores. On the other hand, OpinionSumm represents the higher weighted sentiment component of *DEBSumm*. The results show that *DEBSumm* comprehensively outperforms the state-of-the-art systems. They also show an improvement of 5.2% (ROUGE-1), 7.3% (ROUGE-2) and 5.5% (ROUGE-L) over OpinionSumm. The above results show that sentiment, topic directed or independent of it, is very important factor to compute effective summaries.

Evaluating systems over variable summary size allows us to judge systems over wide range of summary length. Shorter summaries require higher precision and longer summaries require high recall. As the summary size increases, number of sentences which add novel relevant information decreases. Thus, rate of change in scores is not significant. However, in our graph (Figure 1) we find that there is a slight decrease in scores of OpinionSumm and *DEBSumm* from 500 to 1000 words. We believe the reason of such a behavior to inclusion of new noisy data as compared to relevant data. This suggests that more relevance should be given to structural and document features over features representing sentiments. Overall, Figure 1 shows that *DEBSumm* consistently outperforms other systems over different summary sizes.

[5]Each user discussion is considered as a single document while calculating tf-idf values

[6]Note that OpinionSumm is the name given to this system to refer it, throughout, this paper only.

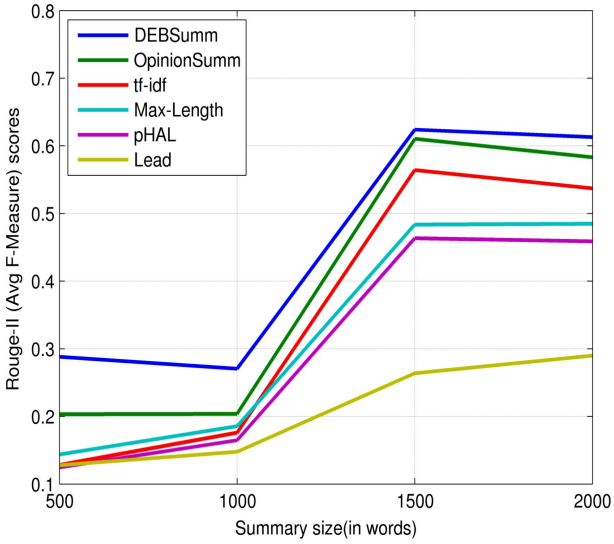


Figure 1: ROUGE-2 (Average F-measure) scores v/s Summary Size (in words)

6. CONCLUSION AND FUTURE WORK

Online debates provides topic related information as well users' opinion regarding the debate topics. Because of large amount of differing opinions, it is necessary to summarize these debates. This paper focuses on summarizing the online debates on the basis of topic directed sentiment and topic related information rich features. To our knowledge, this is the first work in the area of debate summarization. Our sentence ranking based approach ranks debate sentences based on the features related to topic information, sentiment and document statistics. Topic directed sentiment analysis is used to capture sentiment directed towards debate topics whereas topic based co-occurrence measure is used to describe debate topics and sentence closeness. Features commonly used in general text summarization approaches like tf-idf, sentence length and position are also used to capture document statistics based rich sentences. We have compared our system's performance with generic and opinion based state-of-the-art systems. The results show that our system beats all these systems comprehensively.

In this approach, we are averaging sentiment scores of all senses of a word, because of poor state of word sense disambiguation in current scenario, which will not work in all cases. Some words carry different sentiment in different domains for example, 'refined' word is good for 'oil products' whereas bad in the domain of 'agriculture products'. Therefore, next we will be using word sense disambiguation and domain specific sentiment analysis in our system. We will also include debate structure features. These features can leverage DAs occurring along with a high scoring DA. They can also identify related DAs spanned across different users and help in identifying relevant DAs more effectively.

Creating users' profile by capturing their intentions, support by other users and rebuttal arguments can prove a crucial factor in terms of determining the users' expertise. Therefore, we plan to investigate the role of opinion holder in the task of debate summarization.

7. REFERENCES

[1] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau. Sentiment analysis of twitter data. pages 30–38. LSM, 2011.

[2] P. Anand, M. Walker, R. Abbott, J. Tree, R. Bowmani, and M. Minor. Cats rule and dogs drool!: Classifying stance in online debate. In WASSA 2011, pages 1–9, 2011.

[3] C. Aone, M. E. Okurowski, and J. Gorlinsky. Trainable, scalable summarization using robust nlp and machine learning. In COLING, 1998, pages 62–66.

[4] S. Baccianella, A. Esuli, and F. Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. LREC, Valletta, Malta, 2010. ELRA, 2010.

[5] E. Cambria, B. Schuller, Y. Xia, and C. Havasi. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems 28(2)*, pages 15–21, 2013.

[6] E. Cambria, Y. Song, H. Wang, and N. Howard. Semantic multi-dimensional scaling for open-domain sentiment analysis. *IEEE Intelligent Systems*, 2013.

[7] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In SIGIR, 1998, pages 335–336.

[8] M. De Marneffe, B. MacCartney, and C. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006.

[9] H. Edmundson. New methods in automatic extracting. *Journal of the ACM (JACM)*, 16(2):264–285, 1969.

[10] C. Fellbaum. *WordNet*. Springer, 2010.

[11] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In ACL, 2005, pages 363–370.

[12] D. Gillick, K. Riedhammer, B. Favre, and D. Hakkani-Tur. A global optimization framework for meeting summarization. In ICASSP, 2009, pages 4769–4772.

[13] M. Hu and B. Liu. Mining and summarizing customer reviews. In SIGKDD, 2004, pages 168–177.

[14] J. Jagadeesh, P. Pingali, and V. Varma. A relevance-based language modeling approach to duc 2005. In HLT-EMNLP, Vancouver, Canada, 2005.

[15] L.-W. Ku, Y.-T. Liang, and H.-H. Chen. Opinion extraction, summarization and tracking in news and blog corpora. AAAI, 2006, volume 2001.

[16] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In SIGIR, 1995, pages 68–73.

[17] H. Lee, Y. Peirsman, A. Chang, N. Chambers, M. Surdeanu, and D. Jurafsky. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. CONLL : Shared Task, ACL, 2011, pages 28–34.

[18] C. Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, 2004.

[19] H. Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[20] R. Mihalcea. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In ACL, 2004, page 20.

[21] S. Mukherjee and P. Bhattacharyya. Feature specific sentiment analysis for product reviews. CICLING, 2012, pages 475–487.

[22] V. Ng, S. Dasgupta, and S. Arifin. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In COLING-ACL, 2006, pages 611–618.

[23] N. O'Hare, M. Davy, A. Bermingham, P. Ferguson, P. Sheridan, C. Gurrin, and A. F. Smeaton. Topic-dependent sentiment analysis of financial blogs. In CIKM, 2009, pages 9–16.

[24] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. ACL, 2004, page 271.

[25] D. Radev, H. Jing, M. Styś, and D. Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

[26] G. Salton, J. Allan, C. Buckley, and A. Singhal. Automatic analysis, theme generation, and summarization of machine-readable texts. In *Information retrieval and hypertext*, pages 51–73. Springer, 1996.

[27] S. Somasundaran and J. Wiebe. Recognizing stances in online debates. In ACL-IJCNLP of AFNLP, 2009, pages 226–234.

[28] S. Somasundaran and J. Wiebe. Recognizing stances in ideological on-line debates. In NAACL-HLT, 2010, pages 116–124.

[29] D. Wang and Y. Liu. A pilot study of opinion summarization in conversations. In ACL, 2011.

[30] M. Wasson. Using leading text for news summaries: Evaluation results and implications for commercial summarization applications. In COLING-ACL, 1998, pages 1364–1368.

RBEM: A Rule Based Approach to Polarity Detection

Erik Tromp
Adversitement B.V.
Uden, the Netherlands
erik.tromp@adversitement.com

Mykola Pechenizkiy
Department of Computer Science
TU Eindhoven, the Netherlands
m.pechenizkiy@tue.nl

ABSTRACT

We propose the Rule-Based Emission Model (RBEM) algorithm for polarity detection. RBEM uses several kinds of heuristic rules to create an emissive model on polarity patterns. We extensively experiment with our approach on English and Dutch messages extracted from Twitter. Thus we also illustrate that RBEM can be used in multilingual settings and is applicable to social media characterized by use of not always regular language constructs. We demonstrate that designing such an algorithm instead of applying the state-of-the art general purpose classification techniques is a reasonable choice for the automated sentiment classification in practice. Using RBEM we can design a competitive multilingual sentiment classification system showing promising accuracy results of 78.8% on the considered datasets. We provide some further evidence that RBEM-based systems are easy to debug, improve over time and adapt to new application domains.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.5.2 [**Pattern Recognition**]: Design Methodology

General Terms

Design, Algorithms, Performance

Keywords

rule-based polarity detection, emission model, multi-lingual sentiment analysis

1. INTRODUCTION

Sentiment analysis can be performed at different levels of granularity; the document level [19, 11], word level [7] or the sentence or phrase level [17], and with different levels of detail; determining the *polarity* of a message or the emotion expressed [14]. We perform sentiment analysis on social media in which a single message typically consists of one or two sentences. Supported by this observation, the type of granularity we study is the sentence level. We mainly focus on the polarity detection.

Thus, the problem we investigate is how to learn a functional mapping $p = f(m)$ such that given a new message m as input we can output a polarity indication $p \in \{-, +, =\}$ (being negative, positive and neutral respectively) with a high accuracy.

We propose the *Rule-Based Emission Model* (RBEM) algorithm for polarity detection and study it in the context of (multilingual) sentiment analysis on social media. The name of the algorithm indicates the concepts it carries as it uses rules to define an emissive model. Each entity in a message can emit positive or negative sentiment. The rules to determine the polarity of a text are defined on nine different types of patterns.

To show the performance of RBEM in real settings, we consider a three-step approach for designing the automated sentiment analysis. The steps include *language identification*, *part-of-speech tagging* and *polarity detection*, as shown in Figure 1. For language identification we employ LIGA [20].

Figure 1: Multilingual sentiment analysis.

For POS-tagging we use publicly available models for different languages [18].

This approach is generic in a sense that sentiment analysis can be performed on any data source. But it is also easily extensible, allowing to include domain specific knowledge of the particular source, e.g. Twitter hashtags.

We extensively experiment with RBEM for polarity detection as an isolated task and as part of the three-step approach to multi-lingual sentiment classification, showing the utility of RBEM and each preceding step by quantifying the importance of having accurate models in the processing pipeline.

Using RBEM we can design a competitive multilingual sentiment classification system showing promising accuracy results of 78.8% on the considered datasets.

We focus on highlighting the peculiarities of sentiment classification on social media data and argue that designing a focused ruled-based approach instead of applying a state-of-the art general purpose classification techniques is a reasonable choice for this application. Besides benchmark-

WISDOM'13, August 11, 2013, Chicago, USA.

ing, we conduct a case study illustrating the practical utility of RBEM and possibility to continuously improve the performance of polarity detection based on user feedback.

The rest of the paper is organized as follows. We introduce RBEM heuristics, training and classification procedures in Section 2. In Section 3 we summarize the results of the different lines of the experimental evaluation. Section 4 concludes.

2. POLARITY DETECTION WITH RBEM

The rules used in the RBEM algorithm directly stem from nine different *pattern groups*, defined as follows.[1]

- **Positive** patterns are positive when taken out of context. English examples hereof are *good, well done.*
- **Negative** patterns are negative when taken out of context, e.g. *bad, terrible.*
- **Amplifier** patterns strengthen polarity of n entities to their left and right, either positive or negative, e.g. *very much, a lot.*
- **Attenuator** patterns weaken polarity of n entities to their left and right, either positive or negative, e.g. *a little, a tiny bit.*
- **Right Flip** patterns flip the polarity of n entities to their right, e.g. *not, no.*
- **Left Flip** patterns flip the polarity of n entities to their left, e.g. *but, however.*
- **Continuator** patterns continue the emission of polarity, e.g. *and, and also.*
- **Stop** patterns interrupt the emission of polarity. Stop patterns usually are punctuation signs such as a dot or an exclamation mark, expressing the general case that polarity does not cross sentence boundaries.
- **Neutral** patterns do not have any particular meaning but may eliminate the existence of other patterns in a given context.

The need for positive, negative and negation patterns is evident. The need for continuators and left flips has been indicated in [7]: conjunctive words such as *and* usually connect adjectives of the same polarity whereas conjunctive words such as *but* usually connect words of opposing polarity. It is easily seen that certain words strengthen or weaken polarity, these are covered by the amplifier and attenuator patterns. The stop patterns are especially useful in determining sentence-based sentiment as these patterns block polarity emission and typically consist of sentence delimiters such as punctuation. The neutral pattern group does not have a specific logic or rule associated with it but is merely there to eliminate the presence of other patterns when a neutral pattern subsumes a pattern of a different pattern group.

Combining these nine pattern groups using some simple rules allows us to define an emissive model. We next describe how a model is constructed and then define how to classify previously unseen data.

[1]Note that the examples only list words but a pattern can consist of any combination of words and POS-tags. This concept is further explained when we describe how to learn a model.

2.1 Learning RBEM

Each message m of length n is represented as a list $m = [(w_1, t_1), .., (w_n, t_n)]$ of tuples of a word w_i with its respective POS-tag t_i. Upon such a message, patterns can be defined. A pattern is a list of tuples of words and POS-tags represented as m. Patterns belong to a certain pattern group and hence we represent a pattern q as a tuple $q = (g, p)$, where g is the pattern group q belongs to, and p is the list of entities comprising the actual pattern. In general, each element (w'_i, t'_i) of a pattern p consists of a word w'_i which is precisely defined and a POS-tag t'_i which is also precisely defined. As an exception, elements of p may contain wildcards instead. We consider three types of wildcards.

- **Word wildcards** $(\_, t'_i)$**:** in this case we only consider t'_i. w'_i can be any arbitrary word.
- **Single-position wildcards** $(\_, \_)$**:** in this case a single entity can be any arbitrary combination of a single word and a single POS-tag.
- **Multi-position wildcards** $((*, *)$**:** in this case any arbitrary combination of word and POS-tag pairs of any arbitrary length matches the pattern.

Note that word and single-position wildcards can occur at any position in p. But multi-position wildcards can only occur in between two elements that are not multi-position wildcards as co-occurrence of other multi-position wildcards yields another multi-position wildcard.

Our model now simply consists of a set of patterns per pattern group, represented as the set *Model*, containing tuples of groups and patterns; (g, p). All patterns except for the positive and negative patterns adhere to an action radius $\mathcal{E}$. We set $\mathcal{E} = 4$ according to the related experimental results with negation patterns reported in [24]. In general it is possible that the optimal choice of $\mathcal{E}$ may vary from pattern to pattern and/or from one language to the other.

2.2 Classifying with RBEM

When classifying previously unseen data, we perform two steps. First we collect all patterns in our model that match our sentence. Then, we apply a rule associated with each pattern group - with exception of the neutral group - for each pattern present in our message.

Pattern Matching. Each pattern $q = (g, p) \in \mathit{Model}$ is matched against our message $h = [(w_1, t_1), .., (w_n, t_n)]$ where $p = [(v_1, s_1), .., (v_m, s_m)]$. We consider each tuple (w_i, t_i) and evaluate $(v_1, s_1) =_{match} (w_i, t_i)$ where $=_{match}$ is defined as follows:

$$(v_j, s_j) =_{match} (w_i, t_i) \equiv$$

$$\begin{cases} \text{true} & (1) \\ \quad \text{if } j > m, \text{ define } end \leftarrow i & \\ \text{false} & (2) \\ \quad \text{if } i > n & \\ v_j = w_i \wedge s_j = t_i \wedge (v_{j+1}, s_{j+1}) =_{match} (w_{i+1}, t_{i+1}) & (3) \\ \quad \text{if } v_i \neq \_ \wedge v_i \neq * \wedge j \leq m \wedge j \leq n & \\ s_j = t_i \wedge (v_{j+1}, s_{j+1}) =_{match} (w_{i+1}, t_{i+1}) & (4) \\ \quad \text{if } v_i = \_ \wedge s_i \neq \_ \wedge j \leq m \wedge j \leq n & \\ (v_{j+1}, s_{j+1}) =_{match} (w_{i+1}, t_{i+1}) & (5) \\ \quad \text{if } v_i = \_ \wedge s_i = \_ \wedge j \leq m \wedge j \leq n & \\ (v_{j+1}, s_{j+1}) =_{match} (w_{i+1}, t_{i+1}) \vee (v_j, s_j) = & \\ \qquad =_{match} (w_{i+1}, t_{i+1}) & (6) \\ \quad \text{if } v_i = * \wedge j \leq m \wedge j \leq n & \end{cases}$$

Note that in the definition of $=_{match}$, cases (4), (5) and (6) correspond to the three different types of wildcards. Moreover, in the evaluation of the first disjunction of (6), $(v_{j+1}, s_{j+1}) =_{match} (w_{i+1}, t_{i+1})$, it must hold that $v_{j+1} \neq * \wedge s_{j+1} \neq *$ due to the restriction we put on the occurrence of multi-position wildcards.

We match all patterns of all groups against every possible element (w_i, t_i) of m. While doing this, we need to keep track of two positions if a pattern matches; the start position of the match in m and the end position of the match in m. The starting position is i whereas the end position is *end* which is assigned a value in case (1) of $=_{match}$, implying a match between the pattern and the message. We thus get a set of matching patterns containing a start position, an end position and a pattern.

$$matchedPatterns = \{(start, end, (g, [(v_1, s_1), .., (v_n, s_n)])) \mid (v_1, s_1) =_{match} (w_{start}, t_{start})\}$$

Elements of *matchedPatterns* may subsume each other. Subsumption in this sense is defined as follows, where we say that q_1 subsumes q_2 in message m.

$$\exists_{(s_1,e_1,q_1),(s_2,e_2,q_2) \in matchedPatterns} : s_1 \leq s_2 \wedge e_1 \geq e_2 \wedge \neg(s_1 = s_2 \wedge e_1 = e_2) \wedge q_1 \neq q_2$$

All patterns that are subsumed by some other pattern are removed. Note that coinciding patterns, having the same start position as well as the same end position, are not removed but as we deal with sets, such coinciding patterns must be of different pattern groups. Also note that it may be that a pattern containing a wild card may match our sentence multiple times from the same starting position. As the definition of $=_{match}$ dictates, we only find and hence maintain the shortest of such matchings. After removing subsumed patterns, the resulting set *maxPatterns* only contains maximal patterns and is defined as follows. Note that this is where the neutral pattern group plays a role. Whenever a neutral pattern exists in a context that subsumes any other pattern, the neutral pattern is kept whereas the other pattern is discarded. During the application of rules however, nothing is done with this neutral pattern, explaining the name of this pattern group.

$$maxPatterns = \{(s, e, q) | (s, e, q) \in matchedPatterns \wedge \neg(\exists_{(s',e',q') \in matchedPatterns} : s \leq s' \wedge e' \geq e \wedge \neg(s = s' \wedge e = e') \wedge q \neq q')\}$$

Rule Application. After having collected all maximal patterns, we can apply the heuristic rules for each different pattern group, excluding the neutral pattern group. The rules formally work out the motivation for the presence of each pattern group. The order in which the rules are applied is crucial and so is the role of the action radius $\mathcal{E}$. We outline each of the rules in the order in which they are to be applied. We assume we are given a message m and a model $(Model, \mathcal{E})$ on which *maxPatterns* is defined. Every element $e_i = (w_i, t_i) \in m$ has a certain emission value $em(e_i)$ which initially is set to 0 for all $e_i \in m$.

Rule 1. Setting Stops – This rule sets emission boundaries in our message m. It uses all left flip and stop patterns and sets a stop at the starting position of such a pattern. We thus get a set of stops:

$$stops = \{s | (s, f, leftflip) \in maxPatterns \vee (s, f, stop) \in maxPatterns\}$$

Rule 2. Removing Stops – Stops set in the previous step can be removed by continuator patterns. This however, only happens to the left of a continuator pattern. We thus remove all stops that occur closest to the left of a continuator pattern, taking $\mathcal{E}$ into account:

$$stops = stops \setminus \{t | t \in stops \wedge (\exists_{(s,f,continuator) \in maxPatterns} : t \leq s \wedge s - t < \mathcal{E} \wedge \neg(\exists_{t' \in stops} : t < t' \leq s))\}$$

Rule 3. Positive Sentiment Emission – A positive pattern can emit positive sentiment among elements of m. The strength of the emission decays over distance and hence we need a decaying function. We use e^{-x} as decaying function, where x is the distance between the positive pattern and an element of m. The choice of the formula e^{-x} is just a choice made by the authors and is not proven to be the optimal formula. As center for the emission, we take the floor of the center of the pattern in m, computed by taking the center of start and end position. We also need to take all stops into account. For each positive pattern, we update the emission values $em(e_i)$ as follows:

$$\forall_{(s,f,positive) \in maxPatterns} : c = \lfloor \frac{s+f}{2} \rfloor \wedge (\forall_{e_i \in m} : \neg(\exists_{t \in stops} : c \geq i \Rightarrow i \leq t \leq c \vee i \geq c \Rightarrow c \leq t \leq i) \Leftrightarrow em(e_i) = em(e_i) + e^{-i})$$

Rule 4. Negative Sentiment Emission – Negative patterns are dealt with in the same way positive patterns are. The only difference is that our decaying function is now negative, yielding $-e^{-x}$. The updating of emission values happens in the same manner:

$$\forall_{(s,f,negative) \in maxPatterns} : c = \lfloor \frac{s+f}{2} \rfloor \wedge (\forall_{e_i \in m} : \neg(\exists_{t \in stops} : c \geq i \Rightarrow i \leq t \leq c \vee i \geq c \Rightarrow c \leq t \leq i) \Leftrightarrow em(e_i) = em(e_i) + -e^{-i})$$

Rule 5. Amplifying Sentiment – Amplifier patterns amplify sentiment emitted either by positive or negative patterns. Similar to the decaying function used for positive and negative patterns, amplification diminishes over distance. Moreover, since entities may already emit sentiment, we use a multiplicative function instead of an additive function. The function we use is $1 + e^{-x}$ where x is the distance. Again this formula is just chosen by the authors and not proven to be optimal. In contrast to positive and negative patterns, amplifiers adhere to the action radius $\mathcal{E}$. The emission values are updated as follows:

$$\forall_{(s,f,amplifier) \in maxPatterns} : c = \lfloor \frac{s+f}{2} \rfloor \wedge (\forall_{e_i \in m} : (\neg(\exists_{t \in stops} : c \geq i \Rightarrow i \leq t \leq c \vee i \geq c \Rightarrow c \leq t \leq i) \wedge 0 < |c - i| < \mathcal{E}) \Leftrightarrow em(e_i) = em(e_i) \cdot (1 + e^{-i}))$$

Note the $0 < |c-i| < \mathcal{E}$ clause. This constraint dictates that $|c-i|$ is at least 1 in $1-e^{-|c-i|}$ (which is our $1+e^{-x}$ function), thus avoiding the case that we multiply by 0 (when we allow

$|c-i|=0$, we get $1-e^0=0$) and hence completely remove emission values.

Rule 6. Attenuating Sentiment – Attenuator patterns perform the reverse of amplifier patterns and weaken sentiment. To do so, instead of using $1+e^{-x}$, we use $1-e^{-x}$:

$$\forall_{(s,f,amplifier)\in maxPatterns} : c = \lfloor \frac{s+f}{2} \rfloor \wedge$$
$$(\forall_{e_i \in m} : (\neg(\exists_{t\in stops} : c \geq i \Leftrightarrow i \leq t \leq c \vee i \geq c$$
$$\Leftrightarrow c \leq t \leq i) \wedge 0 < |c-i| < \mathcal{E}) \Leftrightarrow$$
$$em(e_i) = em(e_i) \cdot (1 - e^{-i}))$$

Rule 7. Right Flipping Sentiment – Right flip patterns simply flip the emission of sentiment to their right as follows. If there is a stop at the exact center of our right flip, we disregard it:

$$\forall_{(s,f,rightflip)\in maxPatterns} : c = \lfloor \frac{s+f}{2} \rfloor \wedge (\forall_{e_i \in m} : (\neg(\exists_{t\in stops} :$$
$$c < t \leq i) \wedge |c-i| < \mathcal{E}) \Leftrightarrow em(e_i) = -em(e_i))$$

Rule 8. Left Flipping Sentiment – Left flip patterns mirror the effect of right flip patterns:

$$\forall_{(s,f,leftflip)\in maxPatterns} : c = \lfloor \frac{s+f}{2} \rfloor \wedge (\forall_{e_i \in m} : (\neg(\exists_{t\in stops} :$$
$$i \leq t < c) \wedge |c-i| < \mathcal{E}) \Leftrightarrow em(e_i) = -em(e_i))$$

Once the above rules have been applied in the order given, every element e_i of m has an emission value $em(e_i)$. The final polarity of the message is defined by the sum of all emission values for all elements of m:

$$polarity = \sum_{i=1}^{n} em(e_i)$$

Straightforwardly, we say that m is *positive* (class +) if and only if $polarity > 0$. Likewise, we say that m is *negative* (class −) if and only if $polarity < 0$. Whenever $polarity = 0$, we say that m is *neutral* (class =).

When looking at the rules, it becomes clear that the order is important. Stops need to be set first since the other rules depend on stops. Next positive and negative sentiment need to be defined because amplifying, attenuating and flipping sentiment requires sentiment beforehand. Next the sentiment is amplified and attenuated based on the positive and negative emissions defined before. Finally the flips change the direction of the sentiment.

2.3 Related work

Polarity detection has been studied in different communities and in different application domains. The polarity of adjectives was studied in [7] with the use of different conjunctive words. A comprehensive overview of the performance of different machine leaning approaches on polarity detection were presented in [13, 11, 12]. Typically, polarity detection is solved using supervised learning methods but more recently attention is being paid to unsupervised approaches [10].

Some of the recent works adopt a concept-level approach to sentiment analysis [4], which leverages on common sense knowledge for deconstructing natural language text into sentiments. A notable example is [3], in which a two-level affective common sense reasoning framework is proposed to mimic the integration of conscious and unconscious reasoning for sentiment analysis using data mining techniques.

Other works are those of [17, 24, 23, 22]. In these related works, the authors start from bootstrapping methods to label subjective patterns. In their latest work, both subjectivity and polarity detection is performed and evaluated using these patterns along with high precision rules defined in their earlier works.

The idea of using patterns arises from [24] who label subjective expressions (patterns) in their training data. Nevertheless, in their experiments they limit themselves to matching against single-word expressions. The use of rules stems from a different domain. The Brill tagger [2] for POS-tagging uses rules. We borrow this ideology and apply it to polarity detection. The emission aspect of our RBEM algorithm is related to smoothing which is often applied in different machine learning settings. RBEM has also close resemblance to [16] where different rules and patterns are defined on the top of a full linguistic parser output.

More recently attention is being paid to sentiment analysis on social media. Sentiment analysis on Twitter is researched by [6, 9] who use similar methodologies to construct corpora and analyze Twitter messages to determine their polarity. [8] use opinion mining on Twitter to poll the presidential election of the United States in 2008 and show how using Twitter opinion time series can be used to predict future sentiment.

3. EXPERIMENTAL EVALUATION

The goal of our experimental study is three-fold: 1) to benchmark the performance of the proposed RBEM comparing it against popular classification approaches for polarity detection, 2) to study the multilingual settings and the effect of language identification, and 3) to study the portability of RBEM to different application domains.

3.1 Datasets

The training set for our polarity detection algorithm contains messages in multiple languages, multiple sentiments and multiple domains, stemming from social media. The methodology we use to collect data covering different sentiments is similar to [6, 15], in which smileys are used as noisy labels for sentiment and news messages as noisy indicators of neutral messages. For positive and negative messages we query Twitter just for 30 minutes searching for content with happy smileys such as *:), :-), :D etc.* for another 30 minutes with sad smileys such as *:(, :-(, :'(etc.*. For neutral messages, we extract all messages produced by news instances such as the *BBC*, *CNN* (English) or *EenVandaag* (Dutch). We do this again for 30 minutes. From this mixture of polar and non-polar messages we extract patterns for RBEM by manual labeling using a custom web interface that allowed to do this in a quick manner.

For polarity detection we train language specific models. Moreover, the models use POS tags as features known beforehand. LIGA is used to filter out those messages that are neither in Dutch nor in English when querying Twitter with smileys. (LIGA was trained on the other benchmark we created earlier [20]). All the resulting messages are processed by the POS-tagger to form the extended representation containing the parts of speech features.

To construct the test set we collected random data from Twitter as well and then manually annotated the messages. We first labeled messages on language and kept only Dutch and English ones. We next labeled each message on its polar-

Table 1: The sizes of the training/test sets.

Training set/test set size		
	English	Dutch
Positive	3614/205	1202/262
Negative	3458/200	1504/200
Neutral	4706/454	2099/595
Total	11778/859	4805/1057

Table 2: The number of patterns present in the English and Dutch RBEMs.

Pattern Type	English Count	Dutch Count
Amplifiers	67	53
Attenuators	12	6
Rightflips	39	8
Continuators	10	4
Leftflips	5	2
Negatives	541	364
Positives	308	231
Stops	0	2

ity, being either one of positive, negative or neutral. Finally, we extracted numerous RBEM patterns from each message.

The labeling of the test has been performed by multiple annotators, divided into two groups. The first group consisted of three annotators and focused on extracting Dutch messages only and annotating their polarity only, they did not identify RBEM patterns as for testing merely a polarity label is required. The second group consisted of two annotators and focused on extracting English messages only, followed by the same process as for Dutch. We use messages for Dutch in which at least two out of three annotators agreed upon polarity and for English we use messages for which both annotators agreed upon polarity[2].

The size of the resulting training and test sets is shown in Table 1 whereas the numbers of patterns present in our RBEM model used for all experiments are shown in Table 2.

For studying RBEM portability we conducted a case study with additional datasets which we discuss in the corresponding subsection.

3.2 RBEM Accuracy

We compare RBEM against other popular approaches used for sentiment classification, including Prior Polarity Classifier (PPC), Naive Bayes (NB), AdaBoost (AB) with decision stumps as base classifiers, and Support Vector Machines (SVMs).

We experimented with using four different feature spaces where we use either tokens, POS tags, a combination of both or patterns. We also experimented with using all features or the top 2000, 4000 or 8000 features as ranked by mutual information. The resulting accuracies are given in Table 4.

Even though the accuracy of the SVM approach is close to that of the Naive Bayes approach, the SVM has much higher recall whereas the precisions are also comparable. An SVM approach using all features and patterns performs best among the experimented environments, which was also the case for subjectivity detection. Thus SVM outperforms the Naive Bayes on this dataset.

Table 4 only lists using POS-tags for AdaBoost as we found this to be the best feature set for this approach. Using 50 weak learners yields an accuracy of 72.3%, the best accuracy among the settings and features we experimented with for AdaBoost.

The performance of the Prior Polarity Classifiers (Prior) is better than that of SVM and Naive Bayes and close to the performance of the AdaBoost approach. The SentiWordNet variant (SWN) is the most extensive and performs better in terms of precision and overall accuracy.

The left half of the mid section of Table 4 shows the performance of RBEM. Even when we count the misclassification of polar messages we miss out due to insufficient labeling, the accuracy is already comparable to the highest accuracy for AdaBoost and higher than that of the prior polarity classifiers. When we disregard polar messages for which no patterns are present in our model, we obtain a much higher accuracy of 83.9%.

The precision and especially recall of the RBEM algorithm are much higher than those of other approaches. The RBEM algorithm is thus the most favored approach by the experiments conducted.

To investigate how much we can increase the performance of the RBEM algorithm, we investigate how much more the accuracy increases when we have more patterns in RBEM. In approximately six hours of dedicated labeling we found 81 additional patterns. A linguist however would most likely do this much quicker. We mainly label more on positive and negative patterns as we expect to gain the most with these pattern types. Moreover, as our Dutch model was relatively small with respect to our English model, we focused on Dutch patterns. The scores for our metrics we then get for the RBEM algorithm are shown in Table 3. The percentage of polar messages that we do not manage to find due to insufficient labeling drops from 12.9% to 9.4%. The accuracy increases 72.4% to 74.1% when taking all messages into account. When we leave out the messages we cannot find due to insufficient labeling, our accuracy increases from 83.9% to 84.2%, indicating that the newly labeled patterns not only allow us to classify those messages we could not classify previously but also help correct the classification of messages that we misclassified previously.

Table 3: The performance of RBEM with 81 more patterns.

With Missed	A	0.741
	P	0.733
	R	0.876
Without Missed	A	0.842
	P	0.832
	R	0.955
Missed %	9.4%	

3.3 Three-step Process Evaluation

Since we study multilingual settings, we also want to measure the impact of language identification (accuracy) on po-

[2]For the Dutch dataset, the agreement amongst all three annotators is a mere 55%. The agreement between two out of three annotators varies from 65% up to 71%. The agreement on the English dataset is 72.1%.

Table 4: The overall accuracy (A), precision (P) and recall (R) of algorithms for polarity detection. The instances per algorithm having the highest F-measure are shown in bold. The 'Missed %' row lists the fraction of subjective texts not found as such due to insufficient labeling.

Naive Bayes	All	MI 2000	MI 4000	MI 8000
	A 0.616	A 0.504	A 0.506	A 0.511
Tokens	**P 0.551**	P 0.459	P 0.460	P 0.467
	R 0.393	R 0.297	R 0.301	R 0.313
	A 0.585	A 0.491	A 0.493	A 0.482
Tags	P 0.543	P 0.439	P 0.440	P 0.427
	R 0.386	R 0.281	R 0.287	R 0.259
	A 0.589	A 0.495	A 0.494	A 0.502
Mixture	P 0.546	P 0.443	P 0.441	P 0.453
	R 0.387	R 0.286	R 0.287	R 0.292
	A 0.545	A 0.502	A 0.489	A 0.510
Patterns	P 0.497	P 0.419	P 0.447	P 0.428
	R 0.338	R 0.313	R 0.302	R 0.326

SVM	All	MI 2000	MI 4000	MI 8000
	A 0.656	A 0.504	A 0.507	A 0.420
Tokens	P 0.795	P 0.500	P 0.688	P 0.675
	R 0.431	R 0.013	R 0.023	R 0.478
	A 0.451	A 0.551	A 0.531	A 0.544
Tags	P 0.559	P 0.549	P 0.523	P 0.533
	R 0.532	R 0.490	R 0.473	R 0.452
	A 0.654	A 0.540	A 0.547	A 0.564
Mixture	P 0.699	P 0.542	P 0.557	P 0.555
	R 0.486	R 0.496	R 0.499	R 0.464
	A 0.637	A 0.500	A 0.503	A 0.514
Patterns	**P 0.646**	P 0.500	P 0.542	P 0.575
	R 0.564	R 0.131	R 0.261	R 0.279

	RBEM	Prior, Tok	Prior, Tok+Tags	Prior, SWN
	A 0.724	A 0.602	A 0.552	A 0.681
With missed	P 0.719	P 0.611	P 0.577	P 0.737
	R 0.862	R 0.583	R 0.550	R 0.498
	A 0.839	**A 0.769**	A 0.778	A 0.687
Without missed	**P 0.828**	**P 0.785**	P 0.831	P 0.741
	R 0.953	**R 0.747**	R 0.664	R 0.712
Missed %	12.9%	21.4%	28.7%	0.9%

AdaBoost	25 models	50 models	75 models	100 models	250 models
	A 0.664	A 0.691	A 0.685	A 0.678	A 0.698
Tokens	P 0.667	P 0.707	P 0.692	P 0.686	P 0.706
	R 0.638	R 0.654	R 0.649	R 0.644	R 0.658
	A 0.699	**A 0.723**	A 0.715	A 0.703	A 0.691
Tags	P 0.704	**P 0.729**	P 0.722	P 0.709	P 0.697
	R 0.668	**R 0.691**	R 0.685	R 0.677	R 0.666

Table 5: The correctness scores after each step (vertical) for leaving out each possible step (horizontal).

	No LI	No POS	**All included**
Acc. of LI	–	0.971	0.971
Acc. of Pol	0.782	0.795	0.839
Acc. of Complete	0.782	0.778	**0.788**

larity detection.

- **Leaving out Language Identification.** When we do not know the language of a message, we can no longer use language-specific models and hence need to apply more generic models. We thus need to combine the language-specific models into one. For POS-tagging we simply apply all models that are present and use the model showing the highest probabilities of being correct. For polarity detection we apply both the English as well as the Dutch model on the message, sum up the scores for both languages and assign the resulting class.
- **Leaving out POS-tagging.** Polarity detection uses POS-tags as features. When we do not have these tags we need to resort to using tokens only. For polarity detection with RBEM, we use our patterns without regarding the POS-tags.

Table 5 shows the accuracies on the test set for all different scenarios. The columns list the steps left out whereas the rows list the accuracies after each step. The accuracy for a single step is computed by dividing the number of messages correctly classified by that step by the number of messages correctly classified up to that step. The last row indicates the proportion of messages that are polar but which our polarity detection step could not classify as such due to insufficiently labeled data. The *complete* row lists the accuracies on the entire test set, computed by dividing the number of messages correctly classified by all steps by the corpus' size. As POS-tagging is merely used to expand our feature space, we do not evaluate the accuracy after performing this step.

As we compare Table 5 column by column, we observe that the highest accuracy is obtained when all three steps are included. This indicates the importance of each of them.

3.4 RBEM Domain Portability

Use of language is highly dependent upon the domain in which it is being used. As such, it is expected that a generically trained model does not perform as well as it should on a specific domain and that domain-specific models do not port well to other domains. In these experiments we show that regardless of whether a concrete instance of RBEM is domain-agnostic or not, its models are easily adapted to new domains for which no previously annotated data were available.

We illustrate the adaptability of the RBEM algorithm through a real-life use case in which it has been applied to a highly specific domain, being the domain of media and particularly television. We do this by taking the generic base model, its characteristics being given in Table 2, and adapting it to fit the television domain. This process involves human interaction, but we show that the adaptation requires little effort. This is in fierce contrast to general-purpose state-of-the-art classification techniques used for sentiment classification, including e.g. SVMs, supervised sequence embedding [1] or deep learning neural networks [5] with which adaptation of models is a nontrivial labor-intensive process requiring a deep understanding of machine learning.

Experiment setup. To demonstrate the ease of portability to a new domain, we applied the generic model constructed in Section 3.1 to the television domain in two different use cases. For convenience we name them *Experiment 1* and *Experiment 2*. The use cases arose from two real-life scenarios in which two different and non-related Dutch television broadcasters wanted to use our approach for sentiment analysis on social media with respect to specific television shows, news bulletins or movies being broadcasted[3].

Even though language use may be different in the television domain, it is expected that language n-gram characteristics as used by the LIGA algorithm hardly change. This expectation is supported by the experiments conducted in [20] where it is shown that LIGA generalizes well across domains.

Both experiments were conducted in the same manner and both applied solely to Dutch messages originating from Twitter.[4] For both experiments we initially collected data starting from 30 minutes before and 2 hours after a television broadcast exactly once. The data was collected from Twitter by searching for the keywords given in Table 6. For each keyword, the amount of messages extracted is also mentioned.

Each of the Dutch messages (as classified by LIGA) is classified with Dutch RBEM as being *positive*, *neutral* or *negative*. These messages along with their polarity labels have been handed over to domain experts for judgement with the aid of identifying common or drastic patterns that are often misunderstood by the generic RBEM base model. The domain experts were asked to return messages that they identified as being misclassified by RBEM, give their judgement on what the correct label would be and if possible, give a brief argument.

We investigated the received feedback to identify common patterns and drastic misinterpretations by the RBEM algorithm. These common and drastic patterns directly lead to modifications of our base model and can either entail removal of present patterns, addition of new patterns or both.

Generic Model Refinement. For *Experiment 1*, the domain experts returned 90 messages in total across all given keywords that were misclassified according to domain experts. For *Experiment 2* only 8 messages were returned. The great difference in the number of messages returned is not investigated but is most likely to due variance in commitment by the different domain experts.

Table 7 shows the patterns extracted from the resulting messages that corrected the greatest amount of misclassified messages. Note that in these experiments, we did not verify whether these corrections introduce new errors in messages that were not in the set of messages returned by domain

[3]Soap *Goede tijden, slechte tijden*, Talent shows *De beste zangers van Nederland*, Real-life show *Hotter than my daughter*, Game show *Ik hou van Holland*, other shorter names are provided in Table 6.

[4]We intentionally focus on demonstrating domain portability rather than multilingual or source-agnostic aspects; hence the homogeneity of our input data with respect to these two aspects.

Table 6: The keywords used in the portability experiments and the number of resulting messages. Note that for *Experiment 2*, a single message may be included for multiple keywords.

Description	Keyword	# Messages
Experiment 1		
TV show *Babyboom*	#babyboom	226
Talent shows	#bestezangers	199
TV series *Hitch*	#hitch	305
Real-life show	#htmd	969
TV series *House*	#house	199
Game show [5]	#ihvh	772
Experiment 2		
Soap	goede tijden	2465
Soap	goedetijden	432
Soap	gtst	4013
Venue of soap	meerdijk	232

Table 7: Best scoring patterns found in both experiments. G - pattern group (positive +, negative − and neutral =), O - operation (add +, remove -) and #C - number of corrections.

Pattern	G	O	#C
Experiment 1			
$[(huilen, verbpressg)]$	−	−	7
$[(te, partte), (,adj)]$	−	−	3
$[(jammer, verbpressg)]$	−	−	2
$[(jammer, verbpressg), (*, *), (afgelopen, verbpapa)]$	+	+	1
Experiment 2			
$[(goede, adj), (tijden, nounpl)]$	=	+	2
$[(slechte, adj), (tijden, nounpl)]$	=	+	2
$[(stotterd, nounsg)]$	−	+	1

experts.

From *Experiment 1*, we found that the pattern $[(te, partte), (,adj)]$ (in English: *too ...*), which is a negative pattern in the generic base model, expressing that having too much of something is often bad, is not always used to express something negative. Removing this pattern mainly resulted in messages classified as negative before being classified as neutral or even positive after. The words *jammer* (in English: *pity*) and *huilen* (in English: *crying*) are generically associated with negative polarity and hence existed as such in our generic model. In the television domain however, the Dutch word for pity is often used to indicate that it is a pity a show is over and hence is positive instead of negative. Similarly, expressing an emotional act of crying often indicates a television broadcasting has high impact and hence is a positive pattern.

From *Experiment 2*, the main correction was a straightforward one. The television show *goede tijden, slechte tijden* contains the words *goede* and *slechte*, indicating positive and negative sentiment when no context is given. When talking about *goede tijden* in the context of this specific television show however, it is obvious that this is the name of the show and hence bears no emotional value. To this end, adding *goede tijden* (and likewise, *slechte tijden*) as a neutral pattern ensures that when this bigger context is given, the positive pattern containing just the word *goede* is subsumed by this newly introduced neutral pattern and hence eliminated.

After incorporating new patterns based on the feedback by domain experts, we reduced the number of misclassifications from 90 to 32 in *Experiment 1* and from 8 to 1 in *Experiment 2*.

4. CONCLUSIONS

Previous work in the area of sentiment analysis traditionally focused on benchmarking performance of sentiment classification techniques, typically on one language only, usually English as the resources for English are best available. In this paper we introduced a new rule-based approach for polarity detection and investigated its competitive advantages.

The RBEM algorithm provides a solid foundation that is easily extended. Adding more patterns and rules that further increase its accuracy is straightforward when the relation with the other rules is analyzed. Enriching the model via the relevance feedback from the user is also feasible and automating this process is one of the directions of our further work. We will explore methods to find patterns in an automated fashion rather than through a manual labeling process.

Even though many of existing approaches for sentiment analysis can be extended to support multiple languages, this is not a trivial task and typically not included in the studies themselves. We demonstrated the potential of RBEM to be used in a multilingual solution to sentiment analysis by taking both English and Dutch into account.

We targeted multilingual short texts typically present in social media. However, our approach is applicable to sentiment classification in other settings as well.

For our experimental study we constructed two datasets. The training set was constructed by querying Twitter with smileys and scraping news accounts. This yields messages with noisy labels rather than accurate labels. Moreover, this training data is biased as it only contains messages in which smileys are originally present. Constructing more accurate and more representative training data is expected to increase the accuracy of sentiment classification. Additionally, data can be collected for each social medium separately, allowing for more specific, social medium-tailored, models.

In our experimental study we showed the importance of each step, justifying our three-step approach rather than a more generic approach comprising fewer steps. If the sentiment analysis on social media is used for personal use (as envisioned e.g. in [21]) rather than for marketing we can expect that lot of input for RBEM will come through the simple relevance feedback mechanism.

5. REFERENCES

[1] D. Bespalov, Y. Qi, B. Bai, and A. Shokoufandeh. Sentiment classification with supervised sequence encoder. In *Proceedings of European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, volume LNCS 7523, pages 159–174. Springer, 2012.

[2] E. Brill. A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied*

natural language processing (ANCL'92), pages 152–155. Association for Computational Linguistics, 1992.

[3] E. Cambria, D. Olsher, and K. Kwok. Sentic activation: A two-level affective common sense reasoning framework. In *Proceedings of AAAI*, pages 186–192, 2012.

[4] E. Cambria, B. Schuller, Y. Xia, and C. Havasi. New avenues in opinion mining and sentiment analysis. *Intelligent Systems, IEEE*, 28(2):15–21, 2013.

[5] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pages 513–520, 2011.

[6] A. Go, L. Huang, and R. Bhayani. Twitter sentiment analysis using distant supervision.

[7] V. Hatzivassiloglou and K. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the ACL*, pages 174–181, 1997.

[8] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, pages 122–129, 2010.

[9] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 1320–1326, 2010.

[10] G. Paltoglou and M. Thelwall. Twitter, myspace, digg: Unsupervised sentiment analysis in social media. volume 3, pages 66:1–66:19, New York, NY, USA, 2012. ACM.

[11] B. Pang and L. Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, pages 271–278, 2004.

[12] B. Pang and L. Lee. Opinion mining and sentiment analysis. In *Foundations and Trends in Information Retrieval*, 2008.

[13] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of Conference on Empirical methods in natural language processing (EMNLP'02)*, pages 79–86. Association for Computational Linguistics, 2002.

[14] D. Potena and C. Diamantini. Mining opinions on the basis of their affectivity. In *2010 International Symposium on Collaborative Technologies and Systems (CTS)*, pages 245–254, 2010.

[15] J. Read. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *ACLstudent'05 Proceedings of the ACL Student Research Workshop*, pages 43–48, 2005.

[16] R. Remus and C. Hänig. *Towards Well-grounded Phrase-level Polarity Analysis*, pages 380–392. Springer, 2011.

[17] E. Riloff, J. Wiebe, and T. Wilson. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the 7th Conference on Natural Language Learning*, pages 25–32, 2003.

[18] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, 1994.

[19] V. Sindhwani and P. Melville. Document-word co-regularization for semi-supervised sentiment analysis. In *Eighth IEEE International Conference on Data Mining (ICDM'08)*, pages 1025–1030, 2008.

[20] E. Tromp and M. Pechenizkiy. Graph-based n-gram language identification on short texts. In *Proceedings of the 20th Machine Learning conference of Belgium and The Netherlands*, pages 27–34, 2011.

[21] E. Tromp and M. Pechenizkiy. Senticorr: Multilingual sentiment analysis of personal correspondence. In *Proceedings of IEEE ICDM 2011 Workshops*, pages 470–479. IEEE, 2011.

[22] J. Wiebe and R. Micalcea. Word sense and subjectivity. In *Proceedings of ACL'06*, page 1065–1072, 2006.

[23] J. Wiebe, T. Wilson, and C. Cardie. Annotating expressions of opinions and emotions in language. language resources and evaluation. In *Language Resources and Evaluation (formerly Computers and the Humanities)*, pages 347–354. Association for Computational Linguistics, 2005.

[24] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354, 2005.

Cross-lingual Polarity Detection with Machine Translation

Erkin Demirtas
Department of Computer Science
Eindhoven University of Technology
the Netherlands
e.demirtas@student.tue.nl

Mykola Pechenizkiy
Department of Computer Science
Eindhoven University of Technology
the Netherlands
m.pechenizkiy@tue.nl

ABSTRACT

Recent advancements in machine translation foster an interest of its use in sentiment analysis. In this paper, we investigate prospects and limitations of machine translation in sentiment analysis for cross-lingual polarity detection task. We focus on improving classification accuracy in a cross-lingual setting where we have available labeled training instances about particular domain in different languages. We experiment with movie review and product review datasets consisting of polar texts in English and Turkish. The results of the study show that expanding training size with new instances taken from another corpus does not necessarily increase classification accuracy. And this happens primarily not due to (not always accurate) machine translation, but because of the inherent differences in corpora between two subsets written in different languages. Similarly, in case of co-training classification with machine translation we observe from the results that accuracy improvement can be explained by semi-supervised learning with unlabeled data coming from the same domain, but not due to cross-language co-training itself. Our results also show that amount of artificial noise added by machine translation services does not hinder classifiers much in polarity detection task. However, it is important to distinguish the effect of machine translation from the effect of merging different cross-lingual data sources and that like in case of transfer learning we may need to search for ways to account for cross-lingual data distribution differences.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining; I.5.2 [**Pattern Recognition**]: Design Methodology

General Terms

Experimentation, Algorithms, Performance

Keywords

multi-lingual polarity detection, machine translation

WISDOM'13, August 11, 2013, Chicago, USA.

1. INTRODUCTION

Sentiment analysis is an emerging research area which already gathered a lot attention from the NLP community. It studies opinions, sentiments, appraisal, and emotions expressed in text. Recently, rapid growth of digital data and widespread information flow stimulate the development of computational methods in this field.

Although the volume of sentiment analysis research is increasing, majority of studies in the field still concentrates on English. There are different motivations for considering multi-lingual sentiment classification. From an analytics perspective we may want to focus on particular language or compare how much and what positive and negative sentiments are expressed per language of interest [13]. From machine learning perspective and NLP perspectives, multilingual and cross-lingual sentiment classification allows for using language specific models.

Many advanced tools developed for English are not available for other languages yet, which strains the applicability of sentiment analysis on other languages. The main motivation for multi-lingual sentiment analysis is of researchers from different countries want to build sentiment analysis systems in their own languages, but it is more than what it provides for each language at the individual level as it might contribute to our understanding of the global phenomena. Unfortunately the development of complex NLP tools i.e. parsers, taggers, and linguistic resources for each language is very costly and requires expensive human labor. In this regard, the potential of automated machine translation have been studied to leverage its capability, existing sentiment analysis resources and tools available in English to classify sentiments in other languages [3].

Language specific sentiment analysis mostly depends on the monolingual resources and tools that are available for that language. Previous research focus on improving multilingual sentiment analysis resulted in interesting attempts to leverage available resources using machine translation since in most cases they only exist for a limited number of languages.

In this paper, we study whether it is possible to improve classification accuracy in a cross-lingual setting where we have available (labeled) training instances about a particular domain in different languages.

We experiment with movie review and product review datasets consisting of polar texts in English and Turkish. There are already number of publicly available annotated corpora from these domains in English and we also crawled two annotated corpora in Turkish: the first is an annotated

movie review corpus from publicly available website[1], and the second is an annotated multi-domain product review corpus from publicy available e-commerce website[2]. We constructed two benchmark datasets for each corpus that we make publicly available[3] for reproducibility of the experimental study and facilitation of further experimentation by other researchers.

The two goals of the experiments are to investigate 1) whether expanding training size with new machine translated instances taken from another corpus improves classification accuracy for the original corpus and 2) whether co-training with machine translation addresses cross-lingual polarity detection.

It is intuitive that quality of machine translation should be high such that instances from the dataset in the other language do not introduce noise to the training data. However, it is also important to realize that even when we consider the same application domain like movie reviews, due to cultural or other differences, there could be different biases in what people with different background (manifested in use of one or the other language) comment on, like or dislike.

Indeed, the results of the study show that expanding training size with new instances taken from another corpus does not necessarily increase classification accuracy. And this happens primarily not due to (not always accurate) machine translation, but because of the inherent differences in corpora between two subsets written in different languages. Similarly, in case of co-training classification with machine translation we observe from the results that accuracy improvement can be explained by semi-supervised learning with unlabeled data coming from the same domain, but not due to cross-language co-training itself.

Experiment results show that the expansion of our training set improves classification accuracy if we add new instances from same source (also when applying sequentially machine translation from English to Turkish and then back to English). However, when we introduce new training instances from another corpus, i.e. machine translated reviews from Turkish datasets, cross-lingual dissimilarities of two corpora overwhelms positive effects of having a larger training set – the classification accuracy for the target language dataset does not increase.

In our co-training with machine translation experiment we observed an improvement in classifying test data constructed from Turkish movie reviews, i.e. when test instances are in the other language. However, there is no improvement (over co-training iterations) for the reviews in English that is set as our target language. Thus, co-training with use of machine translation likely suffers from the same problem of cross-lingual dissimilarities of two corpora.

Thus, from our experimental study with two distinct applications of machine translation for cross-lingual polarity detection we can conclude two important facts. The quality of current machine translation techniques and services is already sufficient for improving cross-lingual sentiment classification (at least with the general-purpose classification techniques like SVMs). However, it is important to distinguish the effect of machine translation from the effect of merging different cross-lingual data sources and that like in case of transfer learning we may need to search for ways to account for cross-lingual data distribution differences.

The rest of the paper is organized as follows. We discuss related work in Section 2. The settings and the motivation behind this experimental study are explained in Section 3. Section 4 presents the details of the used datasets, experiment setup, and main results. Section 5 concludes with a summary of findings and directions for further research.

2. RELATED WORK

Previously authors developed methods to map sentiment analysis on English to other languages. Mihalcea et al. [10] propose a method to learn multilingual subjective language via cross-language projections. They use the Opinion Finder lexicon [16] and use two bilingual English-Romanian dictionaries to translate the words in the lexicon. Since word ambiguity can appear (Opinion Finder does not mark word senses), they filter as correct translations only the most frequent words. The problem of translating multi-word expressions is solved by translating word-by-word and filtering those translations that occur at least three times on the Web.

Another approach in obtaining subjectivity lexicons for other languages than English was explored by Banea et al. [4]. To this aim, the authors perform three different experiments, obtaining promising results. In the first one, they automatically translate the annotations of the MPQA corpus and thus obtain subjectivity annotated sentences in Romanian. In the second approach, they use the automatically translated entries in the Opinion Finder lexicon to annotate a set of sentences in Romanian. In the last experiment, they reverse the direction of translation and verify the assumption that subjective language can be translated and thus new subjectivity lexicons can be obtained for languages with no such resources.

Brooke et al. [6] experimented with translation from the source (English) to the target language (Spanish) and then used a lexicon-based approach or machine learning for target language document sentiment classification.

Steinberger et al. [12] create sentiment dictionaries in other languages using a method called "triangulation". They translate the data, in parallel, from English and Spanish to other languages and obtain dictionaries from the intersection of these two translations.

Duh et al. [8] presented their opinions about the research of multilingual sentiment classification, and they claimed that domain mismatch was not caused by machine translation (MT) errors, and accuracy degradation would occur even with perfect MT.

Balahur and Turchi [1] employ fully-formed machine translation systems, also study the influence of the difference in translation performance has on the sentiment classification performance. They report even in the worst cases, when the quality of the translated data is not very high, the drop in performance is of maximum 12%.

Similar to our work, Banea et al. [2] report an improvement in classification accuracy when using out-of-language features, yet our work differs from that in couple of major aspects. Our focus is polarity detection, rather than subjectivity analysis which they investigate. Moreover, their training set is only based on the machine translation of an English corpus, and they do not study how to make use of a new dataset from another language in training set.

In our study we investigate the approach for sentiment classification proposed by Wan [15] who constructs a polarity

[1] http://www.beyazperde.com

[2] http://www.hepsiburada.com

[3] The datasets are available at http://www.win.tue.nl/~mpechen/projects/smm/#Datasets

co-training learning system by using the multi-lingual views obtained through the automatic translation of product-reviews into Chinese and English. While [15] provides empirical evidence that leveraging cross-lingual information improves sentiment analysis in Chinese over what could be achieved using monolingual resources alone, it does not provide any results tested on samples taken from English dataset. Thus, as we show in our experimental study, the conclusions from the reported results in [15] should be interpreted with care.

3. CROSS-LINGUAL SENTIMENT CLASSIFICATION

In general if a text is classified as being subjective, we determine whether it expresses a positive or negative opinion. Structured information available in on-line movie reviews helps us in this regard to eliminate neutrality class as we can rely on user's rating associated on his/her review. We can detect polarity of a subjective review, therefore, based on classified instances on beforehand. However, in the real operational settings we would need to have a subjectivity detection mechanism or three-class polarity detection problem formulation for handling neutral messages. To keep the focus we experiment only with polar messages being either positive or negative.

We can consider cross-lingual sentiment classification as a special case of cross-domain classification settings since even two sources from different languages are from same domain they naturally represent different perspective with respect to cultural biases, hidden sentiments etc. We are tempted to explore how much these differences affect classification performance in a set of movie reviews as it may give hints about applicability of cross-domain classification research on cross-lingual sentiment analysis. We also want to see empirical evidences of introduced machine translation noise in sentiment classification and how much it puts a pressure on potential benefits of having a bigger training set which is expanded with machine translated instances.

We consider two distinct machine translation application scenarios. In the first scenario we simply use machine translation to use labeled instances in Turkish for expanding the training set in English considered as the target language for polarity detection.

In the second scenario we consider the co-training approach as viable alternative to leverage machine translated data as it was proposed in [15]. Although we construct labeled Turkish movie and product reviews during our research, for the co-training approach we regard those reviews as unlabeled to be able to setup the similar experimental settings (yet allowing for expanding the evaluation scenarios) and compare our findings with results reported in [15].

We consider the datasets and experiment setup for two scenarios in the following section.

4. EXPERIMENTAL STUDY

4.1 The benchmark

The following datasets are used in the experiments:

English movie reviews[4]: We use the sentence polarity data which was first introduced by [11]. This data consists of 5331 positive and 5331 negative snippets each containing roughly one single sentence. Reviews are gathered from Rotten Tomatoes web pages for movies released in 2002. They classify reviews marked with *fresh* are positive, and those marked with *rotten* are negative.

English multi domain product reviews[5]: This dataset was first introduced by [5]. It contains product reviews taken from Amazon.com from many product types. For our experiment we use a benchmark dataset which they constructed from four categories (books, dvd, electronics, and kitchen appliances) each consisting of 1000 positive and 1000 negative reviews.

Turkish movie reviews: We collect Turkish movie review dataset from Beyazperde web pages. In order to reach same size with the English dataset we restrict this dataset with 5331 positive and 5331 negative sentences. In this website, reviews are marked in scale from 0 to 5 by the same users who made the reviews. We consider a review positive if its rating is equal to or above 4, and negative if it is below or equal to 2.

Turkish multi domain product reviews: After building Turkish movie reviews dataset, we also collect Turkish product reviews from Hepsiburada.com (an online retailer operating in Turkey) to conduct our training set expansion experiment with reviews from different domains. We constructed another benchmark dataset also consisting reviews from books, dvd, electronics, and kitchen appliances categories to use them along with English product reviews. In this website, reviews are marked in scale from 1 to 5, and majority class of reviews converges to 5, that's why we have to consider a small amount of reviews marked with 3 stars as bearing a negative sentiment to be able to construct a balanced set of positive and negative reviews. It has 700 positive and 700 negative reviews for each of the four categories in which average rating of negative reviews is 2.27 and of positive reviews is 4.5.

For each experiment, part of these sets is used in training and evaluation phase, while the test set is always blind to the training phase. A small summary of the four dataset described above provided in Table 1. We explain in following sections how we use these datasets in our experiments.

4.2 Expanding training set with machine translated instances

A number of approaches have been proposed for polarity detection, including Prior Polarity classification (also with use of an opinion lexicon such as SentiWordNet[6], WordNet-Affect[7] or SenticNet[8]), statistical methods such as support vector machines, neural networks, and Naive Bayes among others. Aspect-based methods are introduced to spot more accurate sentiments on entities and their aspects. New approaches relying on semantic relationships in natural language concepts are also investigated under the concept-level sentiment analysis [7]. In our study we use three popular general purpose classification techniques; Naive Bayes, Support Vector Machines (Linear SVC), and Maximum Entropy (MaxEnt) classification.

As we have labeled datasets in English and Turkish, we can immediately apply any of the supervised learning approaches to build monolingual sentiment classifiers for both

[4]The dataset is available at `http://www.cs.cornell.edu/people/pabo/movie-review-data/`

[5]The dataset is available at `http://www.cs.jhu.edu/~mdredze/datasets/sentiment/`

[6]`http://sentiwordnet.isti.cnr.it/`

[7]`http://wndomains.fbk.eu/wnaffect.html`

[8]`http://sentic.net/downloads/`

Table 1: The summary of the datasets used in the experimental study.

	English Movie Reviews	Turkish Movie Reviews	English Product Reviews				Turkish Product Reviews			
			Books	DVD	Electronics	Kitchen Appliances	Books	DVD	Electronics	Kitchen Appliances
Positive	5331	5331	1000	1000	1000	1000	700	700	700	700
Negative	5331	5331	1000	1000	1000	1000	700	700	700	700

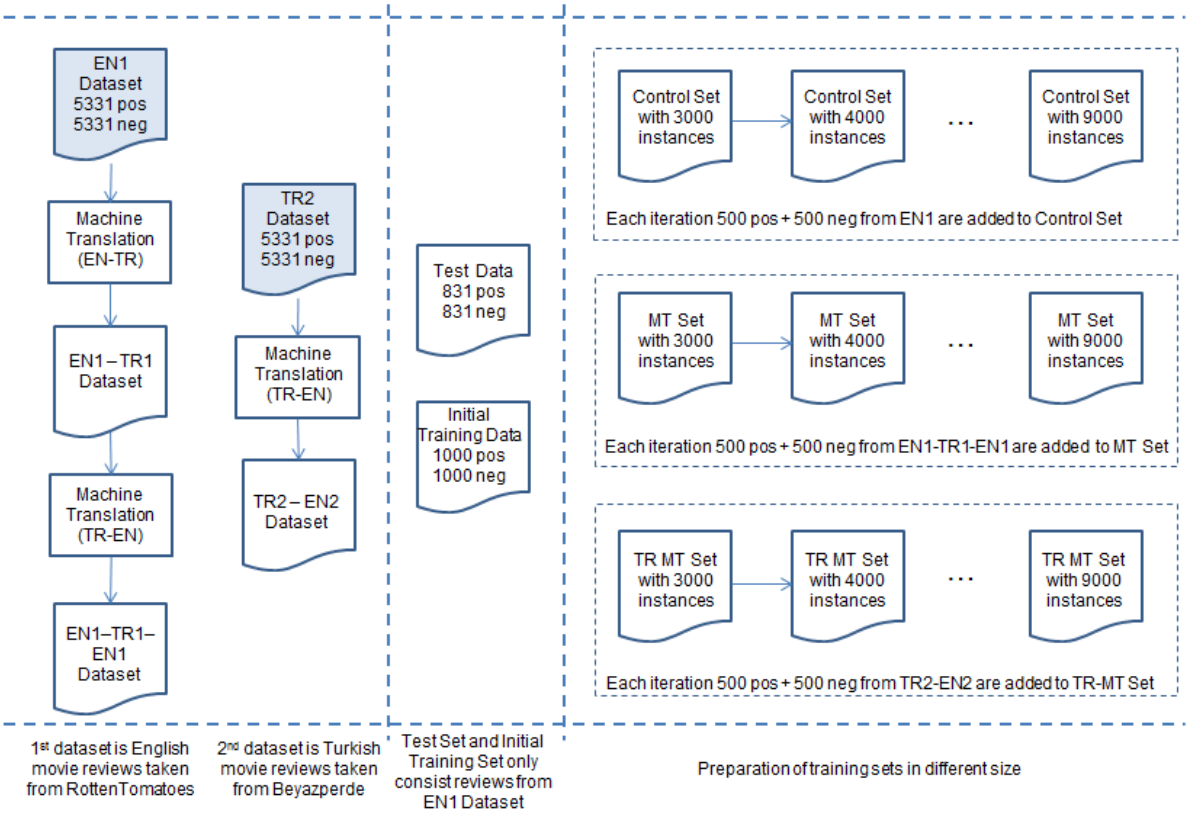


Figure 1: Study of training set expansion with machine machine.

languages. At this point, however, we can also investigate a way of improving classification accuracy of a monolingual classifier for the target language using annotated sources in different languages together. Previously a special case of this question was studied in [2], i.e. a pseudo parallel corpora constructed by machine translation services was used, and the focus was on subjectivity analysis. Their study suggested that the subjectivity classification accuracy can be increased by using features drawn from multiple languages. Our first experiment setting follows the idea of using multiple corpora in different languages but in a more generic way as we do not restrict these corpora to be parallel.

For this experiment, we prepare three types of training sets named as *control*, *machine translated*, and *Turkish machine translated* sets. The control set consists of only reviews from the English dataset. In order to measure the effect of machine translation (quality) we construct machine translated set which consists of reviews from English dataset as well, but then they first translated to Turkish and again back to English just to add artificial translation noise to their original form. Finally, we prepare Turkish machine translated set by compiling reviews from Turkish dataset which are translated to English. For all machine translation processes we use Google Translate service.

As Figure 1 shows, we first sample 1000 (400)[9] positive and 1000 (400) negative reviews from English movie reviews dataset to run the first iteration of the experiment for both training sets. Then, in every next iteration we increase the size of three training sets by adding 500 (100) positive and 500 (100) negative reviews taken from their respective sources. The test set is constructed from 831 (200) positive and 831 (200) negative English reviews that are never used in the training phase.

4.3 Co-training with machine translation

In [15] Wan proposed an application of the co-training method to make use of some amount of unlabeled Chinese product reviews to improve classification accuracy. For our second application scenario while preserving his main idea, we adopt it to our goals. First we use movie reviews instead of product reviews, and we experiment with Turkish-English language setting while Wan uses Chinese-English. These are mostly practical changes in the framework, however, we test combined classifier with reviews taken from both Turkish and English datasets whereas Wan only present results based on test data containing Chinese texts only.

As we can see in Figure 2, training input is the labeled English reviews and some amounts of unlabeled Turkish reviews. The labeled English reviews are translated into labeled Turkish reviews, and the unlabeled Turkish reviews are translated into unlabeled English reviews, by using Google Translate. Therefore, each review is associated with an English version and a Turkish version. The English features and the Turkish features for each review are considered two independent and redundant views of the review.

The co-training Algorithm 1 is then applied to learn two classifiers.

The English and Turkish terms (features) used in our study include unigrams; the feature weight is simply set to term presence following the bag-of-words model. The output value of the Naive Bayes classifier for a review indicates the

[9]numbers in parentheses refer to the setting for product review datasets; without parentheses - to the moview review dataset

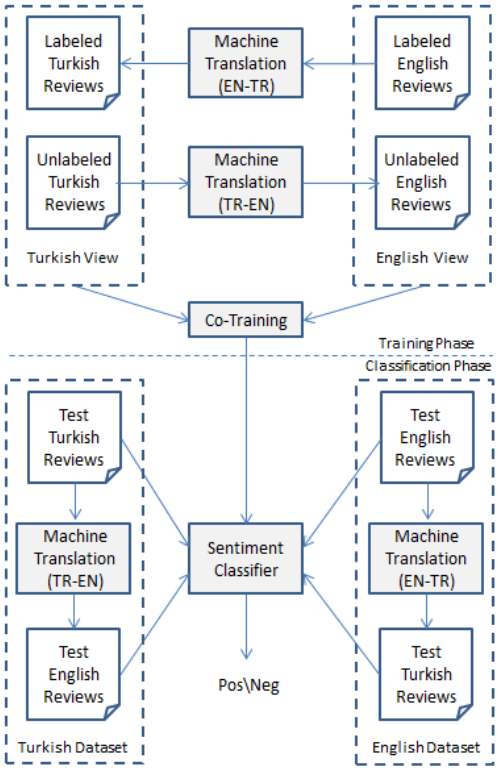


Figure 2: Co-training experiment setup.

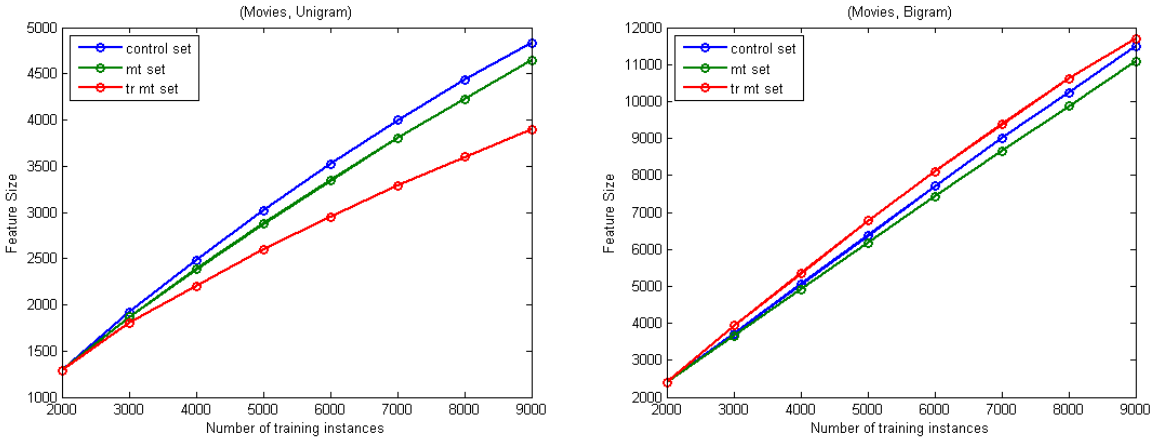
confidence level of the review's classification. In the training phase, the co-training algorithm learns two separate classifiers: C_{en} and C_{tr}. Therefore, in the classification phase, we can obtain two prediction values for a test review, and the average of these values is used as the overall prediction value of the review.

4.4 Results and Discussion

For the training set expansion experiment we present our results in terms of two metrics. First, we measure the feature size increase as we keep adding new instances to the training sets.

The two graphs in Figure 3 show feature size change of movie reviews datasets in which our training sets are represented by unigram and unigram plus bigram features respectively. We observe an interesting behavior of the feature size change in Turkish machine translated set. Despite its slope is smaller in case of unigram feature representation, when we look at bigram representations it produces more features than the any other set does. Relative poor increase in unigram feature size can be explained by the data loss happened during machine translation as such a number of Turkish words could not be translated to English. On the other hand, machine translation introduces some amount of noise as well which portrays itself by producing a vast number of meaningless bigrams.

Accuracy results of the Naive Bayes classifiers on movies reviews datasets are summarized in Figure 4. We can observe some interesting results. First, consistent with our expectation, expanding training size by adding new instances from the same corpus improves the overall accuracy. This



(a) Unigram

(b) Unigram + Bigram

Figure 3: Feature size comparison for the training set expansion experiment.

Algorithm 1 Co-training two classifiers

Input: F_{en} and F_{tr} are redundantly sufficient sets of features, where F_{en} represents the English features, F_{tr} represents the Turkish features, L is a set of labeled training reviews, U is a set of unlabeled reviews
Output: two classifier C_{en} and C_{tr}
for $i \in \{1, 2, \cdots, k\}$ **do**
  Learn the first classifier C_{en} from L based on F_{en}
  Use C_{en} to label reviews from U based on F_{en}
  Choose p positive and n negative the most confidently predicted reviews E_{en} from U
  Learn the second classifier C_{tr} from L based on F_{tr}
  Use C_{tr} to label reviews from U based on F_{tr}
  Choose p positive and n negative the most confidently predicted reviews E_{tr} from U
  Removes reviews $E_{en} \cup F_{tr}$ from U
  Add reviews $E_{en} \cup E_{tr}$ with the corresponding labels to L
end for
return C_{en}, C_{tr}

behavior can be noted following the control set results for both graphs in Figure 4. Machine translation set slightly under-performs than the control set due to the negative effect of machine translation quality, and this difference tends to increase slightly as we add more machine translated sentences to the training set. Nevertheless, the overall effect of machine translation in this case is positive. We can observe 5% increase in accuracy. The results corresponding to the use of Turkish machine translated set (red line fluctuating between 69% and 70%) clearly shows that naive cross-lingual training set expansion does not improve the generalization performance of polarity detection, although we do gather more features from new instances translated from Turkish movie reviews. This problem refers to cross-domain classification as we can regard new features from Turkish reviews as ones from another domain which is not really immediately helpful to classify the test instances taken from the English dataset. These results suggest that an application of resolving cross-corpora dissimilarity may help to utilize labeled instances taken from another language in cross-lingual sentiment analysis.

Table 2: Naïve Bayes classification performance

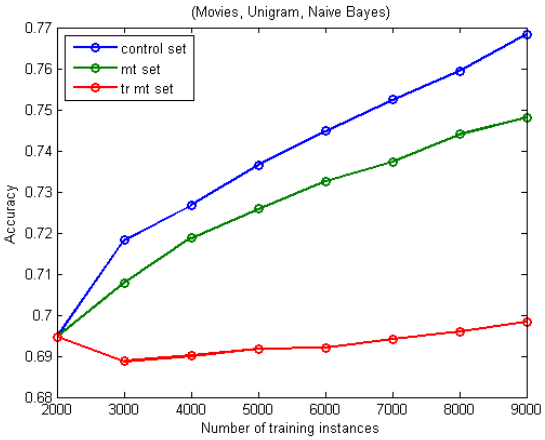
	Initial accuracy	Control set	MT set	TR MT set
Movies	69.5	+10.6	+7.7	+0.5
Books	72.4	+9.2	+8.6	-0.7
DVD	76.0	+4.6	+1.5	-1.1
Electronics	73.0	+8.1	+9.6	-8.6
Kitchen	75.9	+7.2	+8.7	-6.3

Table 3: Linear SVC classification performance

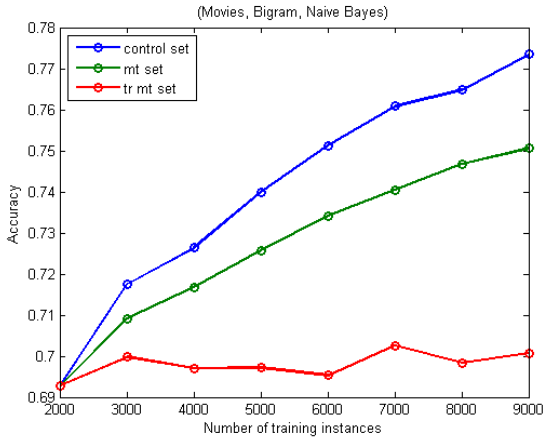
	Initial accuracy	Control set	MT set	TR MT set
Movies	66.0	+11.3	+8.2	+0.5
Books	66.6	+11.1	+14.0	+0.3
DVD	70.3	+7.7	+8.0	-2.7
Electronics	72.4	+7.2	+5.0	-8.0
Kitchen	70.0	+12.3	+11.1	-2.7

This behavior of Naïve Bayes classifier is very similar for Linear SVC and MaxEnt classifiers, and it also generalized to all five datasets we experimented with. The summary of the classification performance is given in Tables 2, 3 and 4. In each table in the first column we give the baseline performance on the initial training data, and the following three columns show the absolute increase (or decrease) in the classification accuracy after the additional training data was added in full according to one of the three setups. We can see from the tables that expanding the training set with additional labeled instances from the same source helps to improve the classification performance and from the different source - does not, and in fact on three datasets even deteriorates the performance.

Co-training experiment results give us insightful details to compare our findings with the ones reported by Wan in [15]. In his paper, Wan evaluates the co-training algorithm by classifying labeled Chinese reviews that are taken from same website and which he used in training phase. We present our

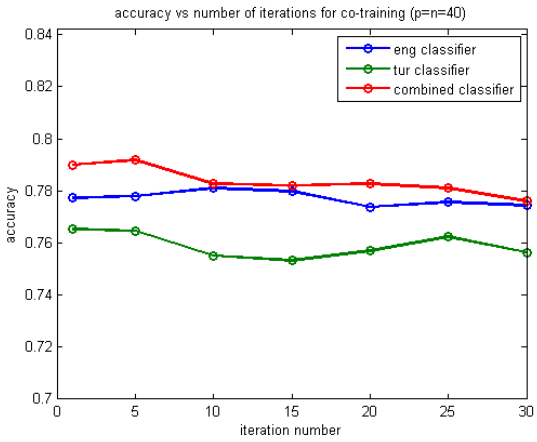


(a) Unigram

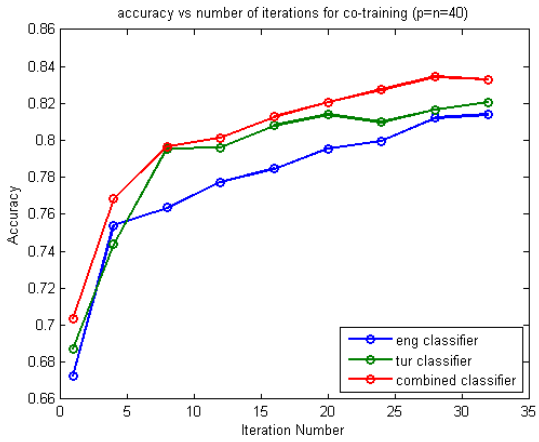


(b) Unigram + Bigram

Figure 4: Generalization accuracies for the training set expansion experiment.



(a) English dataset



(b) Turkish dataset

Figure 5: Accuracy comparison for the co-training experiment.

Table 4: MaxEnt classification performance

	Initial accuracy	Control set	MT set	TR MT set
Movies	68.2	+11.0	+8.8	+0.4
Books	68.7	+12.8	+12.4	+1.8
DVD	71.8	+9.5	+9.6	+1.1
Electronics	74.0	+9.5	+8.0	-7.7
Kitchen	72.4	+12.7	+12.2	-2.2

results based on labeled Turkish movie reviews corresponding to his labeled Chinese reviews, but also the results based on labeled English movie reviews that are discarded from the training phase. Figure 5b confirms findings reported in [15]: tested on labeled Chinese product reviews the combined classifier performs the best and overall accuracy for all classifiers increases in each iteration of co-training. However, co-training approach fails to improve classification accuracy tested on samples from English dataset as we run the algorithm for multiple iteration. For all classifiers (Turkish, English, and combined) we get the highest accuracies with the first iteration that do get better with more iterations. Since proposed co-training approach leverages only unlabeled Chinese reviews (in our work these are replaced by unlabeled Turkish reviews) it resembles semi-supervised learning that aims to increase the classification performance with the aid of some unlabeled data in a language which is the same as the language of the test set. Therefore most of the performance gain presented in [15] is likely due to semi-supervised learning rather than the aid of the English classifier.

5. CONCLUSION AND FUTURE WORK

In this paper, we examined some of the possible improvements in sentiment classification by leveraging labeled or unlabeled data in different languages.

Our experiments show that naive ways of introducing new sources from other languages causes cross-domain dissimilarity issues. This indicates that existing approaches applicable to cross-domain sentiment classification, e.g. [9] and further advancement in this direction might be fruitful for cross-lingual sentiment analysis too. This is one of the directions of our future work.

In this paper we studied how machine translation affects the performance of the general purpose classification techniques. In the future work we plan to consider also techniques specific to sentiment classification like e.g. a rule-based approach to polarity detection [14].

6. REFERENCES

[1] A. Balahur and M. Turchi. Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis. *Computer Speech and Language*, 2013 (In press).

[2] C. Banea, R. Mihalcea, and J. Wiebe. Multilingual subjectivity: are more languages better? In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING'10, pages 28–36, 2010.

[3] C. Banea, R. Mihalcea, and J. Wiebe. *Multilingual Sentiment and Subjectivity Analysis*. Prentice Hall, 2011.

[4] C. Banea, R. Mihalcea, J. Wiebe, and S. Hassan. Multilingual subjectivity analysis using machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 127–135, 2008.

[5] J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, ACL'07, pages 187–205, 2007.

[6] J. Brooke, M. Tofiloski, and M. Taboada. Cross-linguistic sentiment analysis: From english to spanish. In *Proceedings of RANLP'2009*, pages 50–54.

[7] E. Cambria, B. Schuller, Y. Xia, and C. Havasi. New avenues in opinion mining and sentiment analysis. *Intelligent Systems, IEEE*, 28(2):15–21, 2013.

[8] K. Duh, A. Fujino, and M. Nagata. Is machine translation ripe for cross-lingual sentiment classification? In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*, HLT '11, pages 429–433, 2011.

[9] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the Twenty-eight International Conference on Machine Learning (ICML'11)*, volume 27, pages 97–110, June 2011.

[10] R. Mihalcea, C. Banea, and J. Wiebe. Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 976–983, 2007.

[11] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL'05, pages 115–124, 2005.

[12] J. Steinberger, M. Ebrahim, M. Ehrmann, A. Hurriyetoglu, M. Kabadjov, P. Lenkova, R. Steinberger, H. Tanev, S. Vázquez, and V. Zavarella. Creating sentiment dictionaries via triangulation. *Decis. Support Syst.*, 53(4):689–694, 2012.

[13] E. Tromp and M. Pechenizkiy. Senticorr: Multilingual sentiment analysis of personal correspondence. In *Proceedings of IEEE ICDM 2011 Workshops*, pages 470–479. IEEE, 2011.

[14] E. Tromp and M. Pechenizkiy. Rbem: A rule based approach to polarity detection. In *Proceedings of the Workshop on Issues of Sentiment Discovery and Opinion Mining (WISDOM@KDD'13)*. ACM, 2013.

[15] X. Wan. Co-training for cross-lingual sentiment classification. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, ACL'09, pages 235–243, 2009.

[16] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, 2005.

Sentribute: Image Sentiment Analysis from a Mid-level Perspective

Jianbo Yuan
Department of Electrical & Computer Engineering
University of Rochester
Rochester, NY 14627
jyuan10@ece.rochester.edu

Quanzeng You
Department of Computer Science
University of Rochester
Rochester, NY 14627
qyou@cs.rochester.edu

Sean Mcdonough
Department of Electrical & Computer Engineering
University of Rochester
Rochester, NY 14627
smcdono2@u.rochester.edu

Jiebo Luo
Department of Computer Science
University of Rochester
Rochester, NY 14627
jluo@cs.rochester.edu

ABSTRACT

Visual content analysis has always been important yet challenging. Thanks to the popularity of social networks, images become an convenient carrier for information diffusion among online users. To understand the diffusion patterns and different aspects of the social images, we need to interpret the images first. Similar to textual content, images also carry different levels of sentiment to their viewers. However, different from text, where sentiment analysis can use easily accessible semantic and context information, how to extract and interpret the sentiment of an image remains quite challenging. In this paper, we propose an image sentiment prediction framework, which leverages the mid-level attributes of an image to predict its sentiment. This makes the sentiment classification results more interpretable than directly using the low-level features of an image. To obtain a better performance on images containing faces, we introduce eigenface-based facial expression detection as an additional mid-level attributes. An empirical study of the proposed framework shows improved performance in terms of prediction accuracy. More importantly, by inspecting the prediction results, we are able to discover interesting relationships between mid-level attribute and image sentiment.

Categories and Subject Descriptors

H.2.8 [**Database management**]: Database Applications; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Retrieval; I.5.4 [**Pattern Recognition**]: Applications

WISDOM '13, August 11 2013, Chicago, USA

General Terms

Algorithms, Experimentation, Application

Keywords

Image sentiment, Analysis, Mid-level Attributes, Visual Content

1. INTRODUCTION

Nowadays, social networks such as Twitter and microblog such as Weibo become major platforms of information exchange and communication between users, between which the common information carrier is tweets. A recent study shows that images constitute about 36 percent of all the shared links on Twitter[1], which makes visual data mining an interesting and active area to explore. As an old saying has it, an image is worth a thousand words. Much alike textual content based mining approach, extensive studies have been done regarding aesthetics and emotions in images [3, 8, 28]. In this paper, we are focusing on sentiment analysis based on visual information analysis.

So far analysis of textual information has been well developed in areas including opinion mining [18, 20], human decision making [20], brand monitoring [9], stock market prediction [1], political voting forecasts [18, 25] and intelligence gathering [31]. Figure 1 shows and example of image tweets. In contrast, analysis of visual information covers areas such as image information retrieval [4, 33], aesthetics grading [15] and the progress is relatively behind.

Social networks such as Twitter and microblogs such as Weibo provide billions of pieces of both textual and visual information, making it possible to detect sentiment indicated by both textual and visual data respectively. However, sentiment analysis based on a visual perspective is still in its infancy. With respect to sentiment analysis, much work has been done on textual information based sentiment analysis [18, 20, 29], as well as online sentiment dictionary [5, 24].

[1]http://socialtimes.com/is-the-status-update-dead-36-of-tweets-are-photos-infographic_b103245#.UDLhTK9rHY8.wordpress

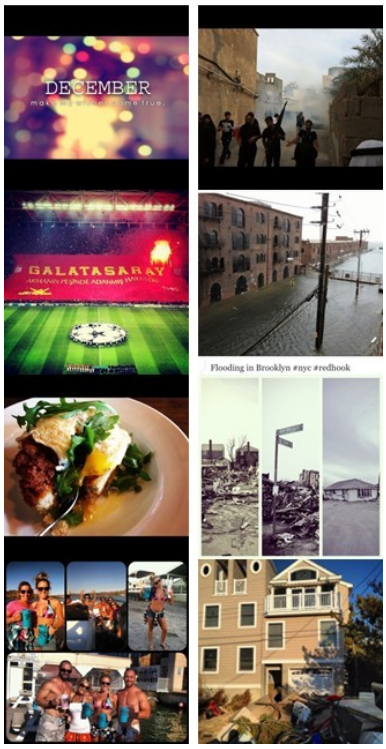


Figure 1: Selected images crawled from Twitter showing (left column) positive sentiment and (right column) negative sentiments.

Semantics and concept learning approaches [6, 19, 16, 22] based on visual features is another way of sentiment analysis without employing textual information. However, semantics and concept learning approaches are hampered by the limitations of object classifier accuracy The analysis of aesthetics [3, 15], interestingness [8] and affect or emotions [10, 14, 17, 32] of images are most related to sentiment analysis based on visual content. Aiming to conduct visual content based sentiment analysis, current approaches includes employing low-level features [10, 11, 12], via facial expression detection [27] and user intent [7]. Sentiment analysis approaches based on low-level features has the limitation of low interpretability, which in turn makes it undesirable for high-level use. Metadata of images is another source of information for high-level feature learning [2]. However, not all images contain such kind of data. Therefore, we proposed Sentribute, an image sentiment analysis algorithm based on mid-level features.

Compared to the state-of-the-art algorithms, our main contribution to this area is two-fold: first, we propose Sentribute, an image-sentiment analysis algorithm based on 102 mid-level attributes, of which results are easier to interpret and ready-to-use for high-level understanding. Second, we introduce eigenface to facial sentiment recognition as a solution for sentiment analysis on images containing people. This is simple but powerful, especially in cases of extreme facial expressions, and contributed an 18% gain in accuracy over decision making only based on mid-level attributes, and 30% over the state of art methods based on low level features.

The remainder of this paper is organized as follows: in Section 2, we present an overview of our proposed Sentribute framework. Section 3 provides details for Sentribute, including low-level feature extraction, mid-level attribute generation, image sentiment prediction, and decision correction based on facial sentiment recognition. Then in Section 4, we test our algorithm on 810 images crawled from Twitter and make a comparison with the state of the art method, which makes prediction based on low-level features and textual information only. Finally, we summarize our findings and possible future extensions of our current work in Section 5.

2. FRAMEWORK OVERVIEW

Figure 2 presents our proposed Sentribute framework. The idea for this algorithm is as follows: first of all, we extract scene descriptor low-level features from the SUN Database [7] and use these four features to train our classifiers by Liblinear [10] for generating 102 predefined mid-level attributes, and then use these attributes to predict sentiments. Meanwhile, facial sentiments are predicted using eigenfaces. This method generates really good results especially in cases of predicting strong positive and negative sentiments, which makes it possible to combine these two predictions and generate a better result for predicting image sentiments with faces. To illustrate how facial sentiment help refine our prediction based on only mid-level attributes, we present an example in Section 4, of how to correct our false positive/negative prediction based on facial sentiment recognition.

3. SENTRIBUTE

In this section we outline the design and construction of the proposed Sentribute, a novel image sentiment prediction method based on mid-level attributes, together with a decision refine mechanism for images containing people. For image sentiment analysis, we conclude the procedure starting from dataset introduction, low-level feature selection, building mid-level attribute classifier, image sentiment prediction. As for facial sentiment recognition, we introduce eigenface to fulfill our intention.

3.1 Dataset

Our proposed algorithm mainly contains three steps: first is to generate mid-level attributes labels. For this part, we train our classifier using SUN Database[2], the first large-scale scene attribute database, initially designed for high-level scene understanding and fine-grained scene recognition [21]. This database includes more than 800 categories and 14,340 images, as well as discriminative attributes labeled by crowd-sourced human studies. Attributes labels are presented in form of zero to three votes, of which 0 vote means this image is the least correlated with this attribute, and three votes means the most correlated. Due to this voting mechanism, we have an option of selecting which set of images to be labeled as positive: images with more than one vote, introduced as soft decision (SD), or images with more than two votes, introduced as hard decision (HD).

Second step of our algorithm is to train sentiment predicting classifiers with images crawled from Twitter together

[2]http://groups.csail.mit.edu/vision/SUN/

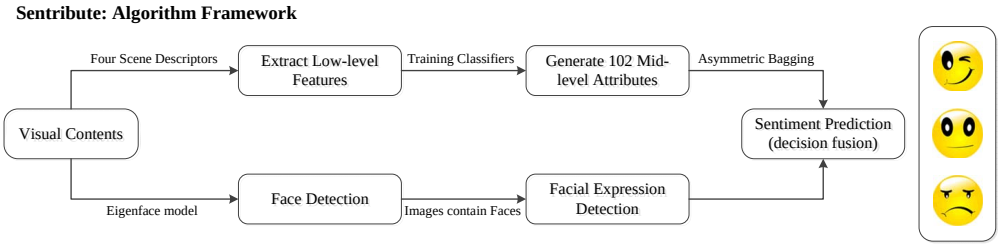


Figure 2: Selected images crawled from Twitter showing (a) positive sentiment and (b) negative sentiments.

with their textual data covering more than 800 images. Twitter is currently one of the most popular microblog platforms. Sentiment ground truth is obtained from visual sentiment ontology[3] with permission of the authors. The dataset includes 1340 positive, 223 negative and 552 neutral image tweets. For testing, we randomly select 810 images, only containing positive (660 tweets) and negative (150 tweets). Figure 1 shows images chosen from our dataset as well as their sentiment labels.

The final step is facial emotion detection for decision fusion mechanism. We chose to use the Karolinska Directed Emotional Faces dataset [13] mainly because the faces are all well aligned with each other and have consistent lighting, which makes generating good eigenface much easier. The dataset contains 70 men and women over two days expressing 7 emotions (scared, anger, disgust, happy, neutral, sad, and surprised) in five different poses (front, left prole, right prole, left angle, right angle).

3.2 Feature Selection

In this section, we are aiming to select low-level features for generating mid-level attributes, and we choose four general scene descriptor: gist descriptor [17], HOG 2x2, self-similarity, and geometric context color histogram features [30]. These four features were chosen because they are each individually powerful and because they can describe distinct visual phenomena in a scene perspective other than using specific object classifier. These scene descriptor features suffer neither from the inconsistent performance compared to commonly used object detectors for high-level semantics analysis of an image, nor from the difficulty of result interpretation generated based on low-level features.

3.3 Generating Mid-level Attribute

Given selected low-level features, we are then able to train our mid-level attribute classifiers based on SUN Database. We have 14,340 dimensions of sampling space, and over 170,000 dimensions of feature space. For classifier options, Liblinear[4] outperforms against LibSVM[5] in cases where the number of samples are huge and the number of feature dimension is huge. Therefore we choose Liblinear toolbox to implement SVM algorithm to achieve time saving.

The selection of mid-level attribute also plays an important part in image sentiment analysis. We choose 102 predefined mid-level attributes based on the following criteria: (1) have descent detection accuracy, (2) potentially correlated to one sentiment label, and (3) easy to interpret. We then select four types of mid-level attributes accordingly: (1) Material: such as metal, vegetation; (2) Function: playing, cooking; (3) Surface property: rusty, glossy; and (4) Spatial Envelope [17]: natural, man-made, enclosed.

We conduct mutual information analysis to discover mid-level attributes that are most correlated with sentiments. For each mid-level attribute, we computed the MI value with respect to both positive and negative sentiment category (Figure 4). Table 1 illustrates 10 most distinguishable mid-level attributes for predicting both positive and negative labels in a descending order based on both SD and HD. Figure 6 demonstrates Average Precision (AP) for the 102 attributes we selected, for both SD and HD. It's not surprising to see that attributes of material (flowers, trees, ice, still water), function (hiking, gaming, competing) and spatial envelop (natural light, congregating, aged/worn) all play an important role according to the result of mutual information analysis

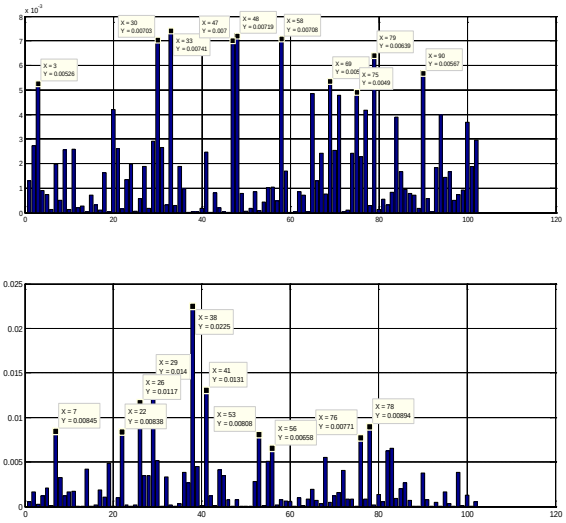


Figure 4: Computing Mutual Information for each label.

3.4 Image Sentiment Prediction

In our dataset we have 660 positive samples and 140 negative samples. It is likely to obtain a biased classifier based on these samples alone. Therefore we introduce asymmetric bagging [23] to dealing with biased dataset. Figure 6 presents the idea of asymmetric bagging: instead of build-

[3] http://visual-sentiment-ontology.appspot.com/
[4] http://www.csie.ntu.edu.tw/~cjlin/liblinear/
[5] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

Figure 3: The images in the table above are grouped by the number of positive labels (votes) received from AMT workers. From left to right the visual presence of each attribute increases [21].

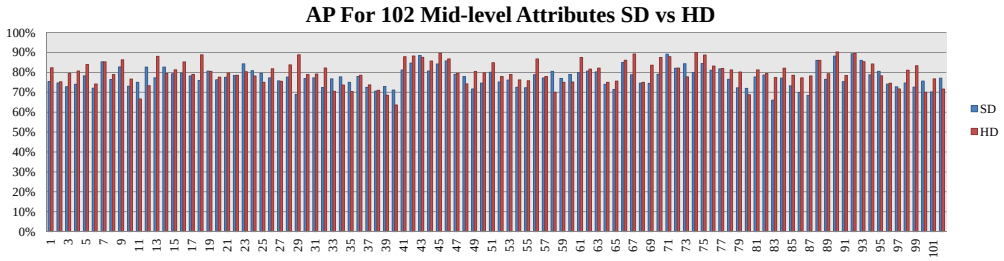


Figure 5: AP of the 102 Attributes based on SD and HD.

Table 1: Attributes with Top 10 Mutual Information.

TOP 10	Soft Decision	Hard Decision
1	congregating	railing
2	flowers	hiking
3	aged/worn	gaming
4	vinyl/linoleum	competing
5	still water	trees
6	natural light	metal
7	glossy	tiles
8	open area	direct sun/sunny
9	glass	aged/worn
10	ice	constructing

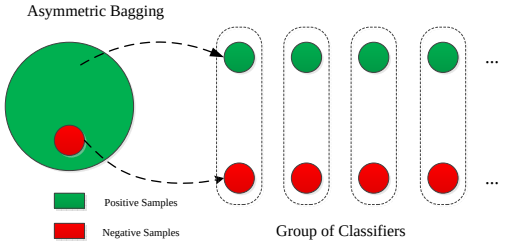


Figure 6: Asymmetric bagging.

ing one classifier, we now build several classifiers, and train them with the same negative samples together with different sampled positive samples of the same amount. Then we can combine their results and build an overall unbiased classifier.

3.5 Facial Sentiment Recognition

Our proposed algorithm, Sentribute, contains a final step of decision fusion mechanism by incorporating eigenface-based emotion detection approach. Images containing faces contribute to a great partition of the whole images that, 382 images from our dataset have faces. Therefore, facial emotion detection is not only useful but important for the overall performance of our algorithm.

In order to recognize emotions from faces we use classes of eigenfaces corresponding to different emotions. Eigenface was one of the earliest successful implementations of facial detection [26]; we modify the algorithm to be suitable for detecting classes of emotions. Though this method is widely appreciated already, we are the first to modify the algorithm to be suitable for detecting classes of emotions, and this method is simple yet surprisingly powerful for detecting facial emotions for front and consistent lightened faces. Note that we are not trying to propose an algorithm that

outperforms the state-of-the-art facial emotion detection algorithms. This is beyond the scope of this paper.

There are seven principal emotions that human's experience: scared (afraid), anger, disgust, happy, neutral, sad, and surprised. Due to the accuracy of the model and the framework of integrating the results with Sentribute, we reduce the set of emotions to positive, neutral, and negative emotions. This is done by classifying the image as one of the seven emotions and then mapping the happy and surprised emotions to positive sentiment, neutral sentiment to itself, and all other emotions to negative sentiment. At a high level, we are computing the eigenfaces for each class of emotion; we then compare the features of these eigenfaces to the features of the target image projected onto the emotion class space.

The algorithm requires a set of faces to train the classifier (more specifically to find the features of the images). We chose to use the Karolinska Directed Emotional Faces dataset [13] for many reasons, specifically the faces are all well aligned with each other and have consistent lighting, which makes generating good eigenfaces much easier. The dataset contains 70 men and women over two days expressing 7 emotions (scared, anger, disgust, happy, neutral, sad, and surprised) in five different poses (front, left profile, right profile, left angle, right angle). We use a subset of the KDEF database for our training set, only using the 7 frontal emotions from one photographing session.

Training the dataset and extracting the eigenfaces from the images of each emotion class was accomplished by using principal component extraction. We preprocess the training data by running it through fdlibmex[6], a fast facial detection algorithm to obtain the position and size of the face. We then extract the face from the general image and scale it to a 64×64 grayscale array; it is then vectored into a 4096 length vector. We concatenate the individual faces from each class into a $M \times N$ array $\mathbf{X}$, where M is the length of each individual image and N is the number of images in the class. We then are able to find the eigenfaces by using Principal Component Extraction. Principal component extraction converts correlated variables, in our case a set of images, into an uncorrelated variables via an orthogonal transform. We implement principal component analysis by first computing the covariance matrix

$$C = (x - \mu)(x - \mu)^T, \tag{1}$$

where μ is the mean of which has been concatenated to the same size of $\mathbf{X}$. The eigenvectors of C are then calculated are arranged by decreasing eigenvalues. Only the twenty largest eigenvectors are chosen for each class of facial emotions. The principle eigenfaces are simply the eigenvectors of the system that have the largest eigenvalues. We compute the features of the class as shown below.

$$\begin{aligned} E^C &= PCA(X^C) \\ F^C &= E^C(X^C - \mu^C) \end{aligned} . \tag{2}$$

In order to classify the target image preprocessing is necessary to preprocess the image as we preprocess the training dataset, which we will denote y. The classification of a test face is performed by comparing the distance of the features of the target face (projected onto the emotion subspace) to the features of the eigenfaces of the subspace. We then choose the class that minimizes this function as the predicted class, specifically

$$\arg\min_C \sum_i \parallel E_i^C(y - \mu^C) - F_i^C \parallel, \tag{3}$$

where i is each individual feature column vector in the array [26].

We then set a threshold value, which was determined empirically, in order to filter out results that are weakly classified. In this case, no result is given. Figure 7 shows examples of classified facial emotions.

(a) Classification: Negative (b) Classification: Positive (c) Classification: Positive

Figure 7: Examples of Eigenface-based Emotion Detection.

4. EXPERIMENTS

4.1 Image Sentiment Prediction

As mentioned before, state-of-the-art sentiment analysis approach can be mainly concluded as: (1) textual information based sentiment analysis, as well as online sentiment dictionary [5, 24] and (2) sentiment analysis based on low-level features. Therefore, in this section, we set three baselines: (1) low-level feature based approach and (2) textual content based approach [24] and (3) online sentiment dictionary SentiStrength [5].

4.1.1 Image Sentiment Classification Performance

First we demonstrate results of our proposed algorithm, image sentiment prediction based on 102 mid-level attributes (SD vs. HD). Both Linear SVM and Logistic Regression algorithms are employed for comparison.

As demonstrated in Table 2, performance of precision for both Linear SVM and Logistic Regression outperforms over that of recall. Owing to the implementation of asymmetric bagging, we are now able to classify negative samples in a decent detection rate. Smaller number of false positive samples and relatively larger number of detected true positive samples contribute to this unbalanced value of precision and recall performance.

Table 2: Image Sentiment Prediction Performance.

		Precision	Recall	Accuracy
Linear SVM	SD	82.6%	56.8%	55.2%
	HD	86.7%	59.1%	61.4%
Logistic Regr	SD	84.3%	54.7%	54.8%
	HD	88.1%	58.8%	61.2%

The next thing we are interested in is the comparison against baseline algorithms.

[6] http://www.mathworks.com/matlabcentral/fileexchange/20976

4.1.2 Low-level Feature Based and Textual Content Based Baselines

For low-level feature based algorithm, Ji et al. employed the following visual features: a dimensional Color Histogram extracted from the RGB color space, a 512 dimensional GIST descriptor [17], a 53 dimensional Local Binary Pattern (LBP), a Bag-of-Words quantized descriptor using a 1000 word dictionary with a 2-layer spatial pyramid, and a 2659 dimensional Classemes descriptor. Both Linear SVM and Logistic Regression algorithms are used for classification. For textual content based algorithm, we choose Contextual Polarity, a phrase level sentiment analysis system [29], as well as SentiStrength API[7]. Table 3 the results of accuracy based on low-level features, mid-level attributes and textual contents.

Table 3: Accuracy of Sentiment Prediction.

(a) Comparison between low-level based algorithm and mid-level based algorithm.

	SVM (low)	Logistic Regr (low)	SVM (mid)
AC	50%	53%	61.4%

(b) Comparison between mid-level visual content based algorithm and textual content based algorithm.

	Contextual Polarity	SentiStrength	SVM (mid)
AC	61.7%	61%	61.4%

4.2 Decision Fusion

The final step of Sentribute is decision fusion. By applying eigenface-based emotion detection, we are able to improve the performance of our decision based on mid-level attributes only. We only take into account images with complete face with reasonable lighting condition. Therefore among all the images with faces, we first employ a face detection process and generate a set of 153 images as the testing data set for facial emotion detection and decision fusion. For each face we detected, we assigned them a label indicating sentiments: 1 for positive, 0 for neutral and -1 for negative sentiments. We thus computed a sentiment score for each image as a whole. For instance, if we detect three faces from an image, two of them are detected as positive and one of them is detected as neutral, then the overall facial sentiment score of this image is 2. These sentiment scores can be used for decision fusion with the decision made based on mid-level attributes only, i.e., we add up the facial sentiment score and the confidence score of the results based on mid-level attributes only returned by our classifiers to implement a decision fusion mechanism. Table 4 shows the improvements in accuracy after decision fusion.

Figure 8 presents examples of TP, FP, TN, FN samples generated by Sentribute. False classified samples show that it's hard to distinguish images only containing texts from both positive and negative labels, and images of big event / cerebration (football game or a concert) from those of protest demonstration. They both share similar general scene descriptors, similar lighting condition, and similar color tone. Another interesting false detected sample is the first image shown in false negative samples. Figures make frown expression on their faces, however the sentiment behind this expression is positive since they were meant to be funny. This sample is initially classified as positive based on mid-level attributes only, and then refined as negative because two strong negative facial expression are detected by our eigenface expression detector. This kind of images shows a better decision fusion metric would be one of our potential improvements.

[7]http://sentistrength.wlv.ac.uk/

Table 4: Accuracy of Sentribute Algorithm.

	Accuracy
Mid-level Based Prediction	64.71%
Facial Emotion Detection	73.86%
Sentribute (After Synthesis)	82.35%

5. CONCLUSION

In this paper we have demonstrated Sentribute, a novel image sentiment prediction algorithm based on mid-level attributes. Asymmetric bagging approach is employed to deal with unbalanced dataset. To enhance our prediction performance, we introduce eigenface-based emotion detection algorithm, which is simple but powerful especially in cases of detecting extreme facial expressions, to dealing with images containing faces and obtain a distinct gain in accuracy over result based on mid-level attributes only. Our proposed algorithm explores current visual content based sentiment analysis approach by employing mid-level attributes and without using textual content. We are aware that this work is just one out of many steps that several potential directions are exciting to set foot on. First, this mid-level based visual content can be introduced to aesthetics analysis as well. Also, a combination of our approach and textual content sentiment analysis approach might be beneficial. Additionally, further application of our proposed work includes but not limited to psychology, public opinion analysis and online activity emotion detection.

Acknowledgments

We thank Professor Shih-Fu Chang's group for providing us with the Columbia University data set for image sentiment analysis.

6. REFERENCES

[1] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[2] E. Cambria and A. Hussain. Sentic album: content-, concept-, and context-based online personal photo management system. *Cognitive Computation*, 4(4):477–496, 2012.

[3] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Studying aesthetics in photographic images using a computational approach. In *Computer Vision–ECCV 2006*, pages 288–301. Springer, 2006.

[4] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (CSUR)*, 40(2):5, 2008.

[5] A. Esuli and F. Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422, 2006.

(a) True positive samples

(b) True negative samples

(c) False positive samples

(d) False negative samples

Figure 8: Examples of Sentiment Detection Results By Sentribute.

[6] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE, 2009.

[7] A. Hanjalic, C. Kofler, and M. Larson. Intent and its discontents: the user at the wheel of the online video search engine. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 1239–1248. ACM, 2012.

[8] P. Isola, J. Xiao, A. Torralba, and A. Oliva. What makes an image memorable? In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 145–152. IEEE, 2011.

[9] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury. Twitter power: Tweets as electronic word of mouth. *Journal of the American society for information science and technology*, 60(11):2169–2188, 2009.

[10] J. Jia, S. Wu, X. Wang, P. Hu, L. Cai, and J. Tang. Can we understand van gogh's mood?: learning to infer affects from images in social networks. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 857–860. ACM, 2012.

[11] P. J. Lang, M. M. Bradley, and B. N. Cuthbert. International affective picture system (iaps): Technical manual and affective ratings, 1999.

[12] B. Li, S. Feng, W. Xiong, and W. Hu. Scaring or pleasing: exploit emotional impact of an image. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 1365–1366. ACM, 2012.

[13] D. Lundqvist, A. Flykt, and A. Öhman. The karolinska directed emotional faces-kdef. cd-rom from department of clinical neuroscience, psychology section, karolinska institutet, stockholm, sweden. Technical report, ISBN 91-630-7164-9, 1998.

[14] J. Machajdik and A. Hanbury. Affective image classification using features inspired by psychology

and art theory. In *Proceedings of the international conference on Multimedia*, pages 83–92. ACM, 2010.

[15] L. Marchesotti, F. Perronnin, D. Larlus, and G. Csurka. Assessing the aesthetic quality of photographs using generic image descriptors. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1784–1791. IEEE, 2011.

[16] M. R. Naphade, C.-Y. Lin, J. R. Smith, B. Tseng, and S. Basu. Learning to annotate video databases. In *SPIE Conference on Storage and Retrieval on Media databases*, 2002.

[17] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[18] B. O'onnor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, pages 122–129, 2010.

[19] V. Ordonez, G. Kulkarni, and T. L. Berg. Im2text: Describing images using 1 million captioned photographs. In *Neural Information Processing Systems (NIPS)*, 2011.

[20] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[21] G. Patterson and J. Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2751–2758. IEEE, 2012.

[22] C. G. Snoek and M. Worring. Concept-based video retrieval. *Foundations and Trends in Information Retrieval*, 2(4):215–322, 2008.

[23] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(7):1088–1099, 2006.

[24] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas. Sentiment strength detection in short informal text. *Journal of the American Society for Information Science and Technology*, 61(12):2544–2558, 2010.

[25] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. In *Proceedings of the fourth international AAAI conference on weblogs and social media*, pages 178–185, 2010.

[26] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 586–591. IEEE, 1991.

[27] V. Vonikakis and S. Winkler. Emotion-based sequence of family photos. In *Proceedings of the 20th ACM international conference on Multimedia*, MM '12, pages 1371–1372, New York, NY, USA, 2012. ACM.

[28] W. Wang and Q. He. A survey on emotional semantic image retrieval. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 117–120. IEEE, 2008.

[29] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354. Association for Computational Linguistics, 2005.

[30] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492. IEEE, 2010.

[31] C. C. Yang and T. D. Ng. Terrorism and crime related weblog social network: Link, content analysis and information visualization. In *Intelligence and Security Informatics, 2007 IEEE*, pages 55–58. IEEE, 2007.

[32] V. Yanulevskaya, J. Uijlings, E. Bruni, A. Sartori, E. Zamboni, F. Bacci, D. Melcher, and N. Sebe. In the eye of the beholder: employing statistical analysis and eye tracking for analyzing abstract paintings. In *Proceedings of the 20th ACM international conference on Multimedia*, MM '12, pages 349–358, New York, NY, USA, 2012. ACM.

[33] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific fusion for image retrieval. In *Computer Vision–ECCV 2012*, pages 660–673. Springer, 2012.