Enhanced Sentiment Learning Using Twitter Hashtags and Smileys

Dmitry Davidov[*1] **Oren Tsur**[*2] **Ari Rappoport**[2]

[1]ICNC / [2]Institute of Computer Science
The Hebrew University
{oren,arir}@cs.huji.ac.il

Abstract

Automated identification of diverse sentiment types can be beneficial for many NLP systems such as review summarization and public media analysis. In some of these systems there is an option of assigning a sentiment value to a single sentence or a very short text.

In this paper we propose a supervised sentiment classification framework which is based on data from Twitter, a popular microblogging service. By utilizing 50 Twitter tags and 15 smileys as sentiment labels, this framework avoids the need for labor intensive manual annotation, allowing identification and classification of diverse sentiment types of short texts. We evaluate the contribution of different feature types for sentiment classification and show that our framework successfully identifies sentiment types of untagged sentences. The quality of the sentiment identification was also confirmed by human judges. We also explore dependencies and overlap between different sentiment types represented by smileys and Twitter hashtags.

1 Introduction

A huge amount of social media including news, forums, product reviews and blogs contain numerous sentiment-based sentences. Sentiment is defined as "a personal belief or judgment that is not founded on proof or certainty"[1]. Sentiment expressions may describe the mood of the writer (happy/sad/bored/grateful/...) or the opinion of the writer towards some specific entity (X is great/I hate X, etc.).

Automated identification of diverse sentiment types can be beneficial for many NLP systems such as review summarization systems, dialogue systems and public media analysis systems. Sometimes it is directly requested by the user to obtain articles or sentences with a certain sentiment value (e.g Give me all positive reviews of product X/ Show me articles which explain why movie X is boring). In some other cases obtaining sentiment value can greatly enhance information extraction tasks like review summarization. While the majority of existing sentiment extraction systems focus on polarity identification (e.g., positive vs. negative reviews) or extraction of a handful of pre-specified mood labels, there are many useful and relatively unexplored sentiment types.

Sentiment extraction systems usually require an extensive set of manually supplied sentiment words or a handcrafted sentiment-specific dataset. With the recent popularity of article tagging, some social media types like blogs allow users to add sentiment tags to articles. This allows to use blogs as a large user-labeled dataset for sentiment learning and identification. However, the set of sentiment tags in most blog platforms is somewhat restricted. Moreover, the assigned tag applies to the whole blog post while a finer grained sentiment extraction is needed (McDonald et al., 2007).

With the recent popularity of the Twitter microblogging service, a huge amount of frequently

[*] Both authors equally contributed to this paper.

[1]WordNet 2.1 definitions.

Coling 2010: Poster Volume, pages 241–249,
Beijing, August 2010

self-standing short textual sentences (*tweets*) became openly available for the research community. Many of these tweets contain a wide variety of user-defined hashtags. Some of these tags are sentiment tags which assign one or more sentiment values to a tweet. In this paper we propose a way to utilize such tagged Twitter data for classification of a wide variety of sentiment types from text.

We utilize 50 Twitter tags and 15 smileys as sentiment labels which allow us to build a classifier for dozens of sentiment types for short textual sentences. In our study we use four different feature types (punctuation, words, n-grams and patterns) for sentiment classification and evaluate the contribution of each feature type for this task. We show that our framework successfully identifies sentiment types of the untagged tweets. We confirm the quality of our algorithm using human judges.

We also explore the dependencies and overlap between different sentiment types represented by smileys and Twitter tags.

Section 2 describes related work. Section 3 details classification features and the algorithm, while Section 4 describes the dataset and labels. Automated and manual evaluation protocols and results are presented in Section 5, followed by a short discussion.

2 Related work

Sentiment analysis tasks typically combine two different tasks: (1) Identifying sentiment expressions, and (2) determining the polarity (sometimes called *valence*) of the expressed sentiment. These tasks are closely related as the purpose of most works is to determine whether a sentence bears a positive or a negative (implicit or explicit) opinion about the target of the sentiment.

Several works (Wiebe, 2000; Turney, 2002; Riloff, 2003; Whitelaw et al., 2005) use lexical resources and decide whether a sentence expresses a sentiment by the presence of lexical items (sentiment words). Others combine additional feature types for this decision (Yu and Hatzivassiloglou, 2003; Kim and Hovy, 2004; Wilson et al., 2005; Bloom et al., 2007; McDonald et al., 2007; Titov and McDonald, 2008a; Melville et al., 2009).

It was suggested that sentiment words may have different senses (Esuli and Sebastiani, 2006; Andreevskaia and Bergler, 2006; Wiebe and Mihalcea, 2006), thus word sense disambiguation can improve sentiment analysis systems (Akkaya et al., 2009). All works mentioned above identify evaluative sentiment expressions and their polarity.

Another line of works aims at identifying a broader range of sentiment classes expressing various emotions such as happiness, sadness, boredom, fear, and gratitude, regardless (or in addition to) positive or negative evaluations. Mihalcea and Liu (2006) derive lists of words and phrases with happiness factor from a corpus of blog posts, where each post is annotated by the blogger with a mood label. Balog et al. (2006) use the mood annotation of blog posts coupled with news data in order to discover the events that drive the dominant moods expressed in blogs. Mishne (2005) used an ontology of over 100 moods assigned to blog posts to classify blog texts according to moods. While (Mishne, 2005) classifies a blog entry (post), (Mihalcea and Liu, 2006) assign a happiness factor to specific words and expressions. Mishne used a much broader range of moods. Strapparava and Mihalcea (2008) classify blog posts and news headlines to six sentiment categories.

While most of the works on sentiment analysis focus on full text, some works address sentiment analysis in the phrasal and sentence level, see (Yu and Hatzivassiloglou, 2003; Wilson et al., 2005; McDonald et al., 2007; Titov and McDonald, 2008a; Titov and McDonald, 2008b; Wilson et al., 2009; Tsur et al., 2010) among others.

Only a few studies analyze the sentiment and polarity of tweets targeted at major brands. Jansen et al. (2009) used a commercial sentiment analyzer as well as a manually labeled corpus. Davidov et al. (2010) analyze the use of the *#sarcasm* hashtag and its contribution to automatic recognition of sarcastic tweets. To the best of our knowledge, there are no works employing Twitter hashtags to learn a wide range of emotions and the relations between the different emotions.

3 Sentiment classification framework

Below we propose a set of classification features and present the algorithm for sentiment classification.

3.1 Classification features

We utilize four basic feature types for sentiment classification: single word features, n-gram features, pattern features and punctuation features. For the classification, all feature types are combined into a single feature vector.

3.1.1 Word-based and n-gram-based features

Each word appearing in a sentence serves as a binary feature with weight equal to the inverted count of this word in the Twitter corpus. We also took each consecutive word sequence containing 2–5 words as a binary n-gram feature using a similar weighting strategy. Thus n-gram features always have a higher weight than features of their component words, and rare words have a higher weight than common words. Words or n-grams appearing in less than 0.5% of the training set sentences do not constitute a feature. ASCII smileys and other punctuation sequences containing two or more consecutive punctuation symbols were used as single-word features. Word features also include the substituted meta-words for URLs, references and hashtags (see Subsection 4.1).

3.1.2 Pattern-based features

Our main feature type is based on surface patterns. For automated extraction of patterns, we followed the pattern definitions given in (Davidov and Rappoport, 2006). We classified words into high-frequency words (HFWs) and content words (CWs). A word whose corpus frequency is more (less) than F_H (F_C) is considered to be a HFW (CW). We estimate word frequency from the training set rather than from an external corpus. Unlike (Davidov and Rappoport, 2006), we consider all single punctuation characters or consecutive sequences of punctuation characters as HFWs. We also consider URL, REF, and HASHTAG tags as HFWs for pattern extraction. We define a pattern as an ordered sequence of high frequency words and slots for content words. Following (Davidov and Rappoport, 2008), the F_H and F_C thresholds were set to 1000 words per million (upper bound for F_C) and 100 words per million (lower bound for F_H)[2].

The patterns allow 2–6 HFWs and 1–5 slots for CWs. To avoid collection of patterns which capture only a part of a meaningful multiword expression, we require patterns to start and to end with a HFW. Thus a minimal pattern is of the form [HFW] [CW slot] [HFW]. For each sentence it is possible to generate dozens of different patterns that may overlap. As with words and n-gram features, we do not treat as features any patterns which appear in less than 0.5% of the training set sentences.

Since each feature vector is based on a single sentence (tweet), we would like to allow approximate pattern matching for enhancement of learning flexibility. The value of a pattern feature is estimated according the one of the following four scenarios[3]:

$$\begin{cases} \frac{1}{count(p)}: & \text{Exact match – all the pattern components appear in the sentence in correct order without any additional words.} \\ \frac{\alpha}{count(p)}: & \text{Sparse match – same as exact match but additional non-matching words can be inserted between pattern components.} \\ \frac{\gamma * n}{N * count(p)}: & \text{Incomplete match – only } n > 1 \text{ of } N \text{ pattern components appear in the sentence, while some non-matching words can be inserted in-between. At least one of the appearing components should be a HFW.} \\ 0: & \text{No match – nothing or only a single pattern component appears in the sentence.} \end{cases}$$

$0 \leq \alpha \leq 1$ and $0 \leq \gamma \leq 1$ are parameters we use to assign reduced scores for imperfect matches. Since the patterns we use are relatively long, exact matches are uncommon, and taking advantage of partial matches allows us to significantly reduce the sparsity of the feature vectors. We used $\alpha = \gamma = 0.1$ in all experiments.

This pattern based framework was proven efficient for sarcasm detection in (Tsur et al., 2010;

[2]Note that the F_H and F_C bounds allow overlap between some HFWs and CWs. See (Davidov and Rappoport, 2008) for a short discussion.

[3]As with word and n-gram features, the maximal feature weight of a pattern p is defined as the inverse count of a pattern in the complete Twitter corpus.

Davidov et al., 2010).

3.1.3 Efficiency of feature selection

Since we avoid selection of textual features which have a training set frequency below 0.5%, we perform feature selection incrementally, on each stage using the frequencies of the features obtained during the previous stages. Thus first we estimate the frequencies of single words in the training set, then we only consider creation of n-grams from single words with sufficient frequency, finally we only consider patterns composed from sufficiently frequent words and n-grams.

3.1.4 Punctuation-based features

In addition to pattern-based features we used the following generic features: (1) Sentence length in words, (2) Number of "!" characters in the sentence, (3) Number of "?" characters in the sentence, (4) Number of quotes in the sentence, and (5) Number of capitalized/all capitals words in the sentence. All these features were normalized by dividing them by the (maximal observed value *times* averaged maximal value of the other feature groups), thus the maximal weight of each of these features is equal to the averaged weight of a single pattern/word/n-gram feature.

3.2 Classification algorithm

In order to assign a sentiment label to new examples in the test set we use a k-nearest neighbors (kNN)-like strategy. We construct a feature vector for each example in the training and the test set. We would like to assign a sentiment class to each example in the test set. For each feature vector V in the test set, we compute the Euclidean distance to each of the matching vectors in the training set, where matching vectors are defined as ones which share at least one pattern/n-gram/word feature with v.

Let $t_i, i = 1 \ldots k$ be the k vectors with lowest Euclidean distance to v[4] with assigned labels $L_i, i = 1 \ldots k$. We calculate the mean distance $d(t_i, v)$ for this set of vectors and drop from the set up to five outliers for which the distance was more then twice the mean distance. The label assigned to v is the label of the majority of the remaining vectors.

If a similar number of remaining vectors have different labels, we assigned to the test vector the most frequent of these labels according to their frequency in the dataset. If there are no matching vectors found for v, we assigned the default "no sentiment" label since there is significantly more non-sentiment sentences than sentiment sentences in Twitter.

4 Twitter dataset and sentiment tags

In our experiments we used an extensive Twitter data collection as training and testing sets. In our training sets we utilize sentiment hashtags and smileys as classification labels. Below we describe this dataset in detail.

4.1 Twitter dataset

We have used a Twitter dataset generously provided to us by Brendan O'Connor. This dataset includes over 475 million tweets comprising roughly 15% of all public, non-"low quality" tweets created from May 2009 to Jan 2010. Tweets are short sentences limited to 140 UTF-8 characters. All non-English tweets and tweets which contain less than 5 proper English words[5] were removed from the dataset.

Apart of simple text, tweets may contain URL addresses, references to other Twitter users (appear as @<user>) or a content tags (also called *hashtags*) assigned by the tweeter (#<tag>) which we use as labels for our supervised classification framework.

Two examples of typical tweets are: *"#ipad #sucks and 6,510 people agree. See more on Ipad sucks page: http://j.mp/4OiYyg?"*, and *"Pay no mind to those who talk behind ur back, it simply means that u're 2 steps ahead. #ihatequotes"*. Note that in the first example the hashtagged words are a grammatical part of the sentence (it becomes meaningless without them) while *#ihatequotes* of the second example is a mere sentiment label and not part of the sentence. Also note that hashtags can be composed of multiple words (with no spaces).

[4] We used $k = 10$ for all experiments.

[5] Identification of proper English words was based on an available WN-based English dictionary

Category	# of tags	% agreement
Strong sentiment	52	87
Likely sentiment	70	66
Context-dependent	110	61
Focused	45	75
No sentiment	3564	99

Table 1: Annotation results (2 judges) for the 3852 most frequent tweeter tags. The second column displays the average number of tags, and the last column shows % of tags annotated similarly by two judges.

During preprocessing, we have replaced URL links, hashtags and references by URL/REF/TAG meta-words. This substitution obviously had some effect on the pattern recognition phase (see Section 3.1.2), however, our algorithm is robust enough to overcome this distortion.

4.2 Hashtag-based sentiment labels

The Twitter dataset contains above 2.5 million different user-defined hashtags. Many tweets include more than a single tag and 3852 "frequent" tags appear in more than 1000 different tweets. Two human judges manually annotated these frequent tags into five different categories: 1 – strong sentiment (e.g *#sucks* in the example above), 2 – most likely sentiment (e.g., *#notcute*), 3 – context-dependent sentiment (e.g., *#shoutsout*), 4 – focused sentiment (e.g., *#tmobilesucks* where the target of the sentiment is part of the hashtag), and 5 – no sentiment (e.g. *#obama*). Table 1 shows annotation results and the percentage of similarly assigned values for each category.

We selected 50 hashtags annotated "1" or "2" by both judges. For each of these tags we automatically sampled 1000 tweets resulting in 50000 labeled tweets. We avoided sampling tweets which include more than one of the sampled hashtags. As a no-sentiment dataset we randomly sampled 10000 tweets with no hashtags/smileys from the whole dataset assuming that such a random sample is unlikely to contain a significant amount of sentiment sentences.

4.3 Smiley-based sentiment labels

While there exist many "official" lists of possible ASCII smileys, most of these smileys are infrequent or not commonly accepted and used as sentiment indicators by online communities. We used the Amazon Mechanical Turk (AMT) service in order to obtain a list of the most commonly used and unambiguous ASCII smileys. We asked each of ten AMT human subjects to provide at least 6 commonly used ASCII mood-indicating smileys together with one or more single-word descriptions of the smiley-related mood state. From the obtained list of smileys we selected a subset of 15 smileys which were (1) provided by at least three human subjects, (2) described by at least two human subject using the same single-word description, and (3) appear at least 1000 times in our Twitter dataset. We then sampled 1000 tweets for each of these smileys, using these smileys as sentiment tags in the sentiment classification framework described in the previous section.

5 Evaluation and Results

The purpose of our evaluation was to learn how well our framework can identify and distinguish between sentiment types defined by tags or smileys and to test if our framework can be successfully used to identify sentiment types in new untagged sentences.

5.1 Evaluation using cross-validation

In the first experiment we evaluated the consistency and quality of sentiment classification using cross-validation over the training set. Fully automated evaluation allowed us to test the performance of our algorithm under several different feature settings: *Pn+W-M-Pt-*, *Pn+W+M-Pt-*, *Pn+W+M+Pt-*, *Pn-W-M-Pt+* and *FULL*, where $+/-$ stands for utilization/omission of the following feature types: *Pn*:punctuation, *W*:Word, *M*:n-grams (M stands for 'multi'), *Pt*:patterns. *FULL* stands for utilization of all feature types.

In this experimental setting, the training set was divided to 10 parts and a 10-fold cross validation test is executed. Each time, we use 9 parts as the labeled training data for feature selection and construction of labeled vectors and the remaining part is used as a test set. The process was repeated ten times. To avoid utilization of labels as strong features in the test set, we removed all instances of involved label hashtags/smileys from the tweets used as the test set.

Setup	Smileys	Hashtags
random	0.06	0.02
Pn+W-M-Pt-	0.16	0.06
Pn+W+M-Pt-	0.25	0.15
Pn+W+M+Pt-	0.29	0.18
Pn-W-M-Pt+	0.5	0.26
FULL	0.64	0.31

Table 2: Multi-class classification results for smileys and hashtags. The table shows averaged harmonic f-score for 10-fold cross validation. 51 (16) sentiment classes were used for hashtags (smileys).

Hashtags	Avg	#hate	#jealous	#cute	#outrageous
Pn+W-M-Pt-	0.57	0.6	0.55	0.63	0.53
Pn+W+M-Pt-	0.64	0.64	0.67	0.66	0.6
Pn+W+M+Pt-	0.69	0.66	0.67	0.69	0.64
Pn-W-M-Pt+	0.73	0.75	0.7	0.69	0.69
FULL	0.8	0.83	0.76	0.71	0.78
Smileys	Avg	:)	;)	$X($	$: d$
Pn+W-M-Pt-	0.64	0.66	0.67	0.56	0.65
Pn+W+M-Pt-	0.7	0.73	0.72	0.64	0.69
Pn+W+M+Pt-	0.7	0.74	0.75	0.66	0.69
Pn-W-M-Pt+	0.75	0.78	0.75	0.68	0.72
FULL	0.86	0.87	0.9	0.74	0.81

Table 3: Binary classification results for smileys and hashtags. *Avg* column shows averaged harmonic f-score for 10-fold cross validation over *all* 50(15) sentiment hashtags (smileys).

Multi-class classification. Under multi-class classification we attempt to assign a single label (51 labels in case of hashtags and 16 labels in case of smileys) to each of vectors in the test set. Note that the random baseline for this task is 0.02 (0.06) for hashtags (smileys). Table 2 shows the performance of our framework for these tasks.

Results are significantly above the random baseline and definitely nontrivial considering the equal class sizes in the test set. While still relatively low (0.31 for hashtags and 0.64 for smileys), we observe much better performance for smileys which is expected due to the lower number of sentiment types.

The relatively low performance of hashtags can be explained by ambiguity of the hashtags and some overlap of sentiments. Examination of classified sentences reveals that many of them can be reasonably assigned to more than one of the available hashtags or smileys. Thus a tweet *"I'm reading stuff that I DON'T understand again! hahaha...wth am I doing"* may reasonably match tags #sarcasm, #damn, #haha, #lol, #humor, #angry etc. Close examination of the incorrectly classified examples also reveals that substantial amount of tweets utilize hashtags to explicitly indicate the specific hashtagged sentiment, in these cases that no sentiment value could be perceived by readers unless indicated explicitly, e.g. *"De Blob game review posted on our blog. #fun"*. Obviously, our framework fails to process such cases and captures noise since no sentiment data is present in the processed text labeled with a specific sentiment label.

Binary classification. In the binary classification experiments, we classified a sentence as either appropriate for a particular tag or as not bearing any sentiment[6]. For each of the 50 (15) labels for hashtags (smileys) we have performed a binary classification when providing as training/test sets only positive examples of the specific sentiment label together with non-sentiment examples. Table 3 shows averaged results for this case and specific results for selected tags. We can see that our framework successfully identifies diverse sentiment types. Obviously the results are much better than those of multi-class classification, and the observed > 0.8 precision confirms the usefulness of the proposed framework for sentiment classification of a variety of different sentiment types.

We can see that even for binary classification settings, classification of smiley-labeled sentences is a substantially easier task compared to classification of hashtag-labeled tweets. Comparing the contributed performance of different feature types we can see that punctuation, word and pattern features, each provide a substantial boost for classification quality while we observe only a marginal boost when adding n-grams as classification features. We can also see that pattern features contribute the performance more than all other features together.

5.2 Evaluation with human judges

In the second set of experiments we evaluated our framework on a test set of unseen and untagged tweets (thus tweets that were not part of the train-

[6]Note that this is a useful application in itself, as a filter that extracts sentiment sentences from a corpus for further focused study/processing.

ing data), comparing its output to tags assigned by human judges. We applied our framework with its FULL setting, learning the sentiment tags from the training set for hashtags and smileys (separately) and executed the framework on the reduced Tweeter dataset (without untagged data) allowing it to identify at least five sentences for each sentiment class.

In order to make the evaluation harsher, we removed all tweets containing at least one of the relevant classification hashtags (or smileys). For each of the resulting 250 sentences for hashtags, and 75 sentences for smileys we generated an 'assignment task'. Each task presents a human judge with a sentence and a list of ten possible hashtags. One tag from this list was provided by our algorithm, 8 other tags were sampled from the remaining 49 (14) available sentiment tags, and the tenth tag is from the list of frequent non-sentiment tags (e.g. *travel* or *obama*). The human judge was requested to select the 0-2 most appropriate tags from the list. Allowing assignment of multiple tags conforms to the observation that even short sentences may express several different sentiment types and to the observation that some of the selected sentiment tags might express similar sentiment types.

We used the Amazon Mechanical Turk service to present the tasks to English-speaking subjects. Each subject was given 50 tasks for Twitter hashtags or 25 questions for smileys. To ensure the quality of assignments, we added to each test five manually selected, clearly sentiment bearing, assignment tasks from the tagged Twitter sentences used in the training set. Each set was presented to four subjects. If a human subject failed to provide the intended "correct" answer to at least two of the control set questions we reject him/her from the calculation. In our evaluation the algorithm is considered to be correct if one of the tags selected by a human judge was also selected by the algorithm. Table 4 shows results for human judgement classification. The agreement score for this task was $\kappa = 0.41$ (we consider agreement when at least one of two selected items are shared).

Table 4 shows that the majority of tags selected by humans matched those selected by the algorithm. Precision of smiley tags is substantially higher than of hashtag labels, due to the lesser number of possible smileys and the lesser ambiguity of smileys in comparison to hashtags.

Setup	% Correct	% No sentiment	Control
Smileys	84%	6%	92%
Hashtags	77%	10%	90%

Table 4: Results of human evaluation. The second column indicates percentage of sentences where judges find no appropriate tags from the list. The third column shows performance on the control set.

Hashtags	#happy	#sad	#crazy	# bored
#sad	0.67	-	-	-
#crazy	0.67	0.25	-	-
#bored	0.05	0.42	0.35	-
#fun	1.21	0.06	1.17	0.43
Smileys	**:)**	**;)**	**: (**	***X(***
;)	3.35	-	-	-
: (	3.12	0.53	-	-
X(	1.74	0.47	2.18	-
: *S*	1.74	0.42	1.4	0.15

Table 5: Percentage of co-appearance of tags in tweeter corpus.

5.3 Exploration of feature dependencies

Our algorithm assigns a single sentiment type for each tweet. However, as discussed above, some sentiment types overlap (e.g., *#awesome* and *#amazing*). Many sentences may express several types of sentiment (e.g., *#fun* and *#scary* in *"Oh My God http://goo.gl/fb/K2N5z #entertainment #fun #pictures #photography #scary #teaparty"*). We would like to estimate such inter-sentiment dependencies and overlap automatically from the labeled data. We use two different methods for overlap estimation: tag co-occurrence and feature overlap.

5.3.1 Tag co-occurrence

Many tweets contain more than a single hashtag or a single smiley type. As mentioned, we exclude such tweets from the training set to reduce ambiguity. However such tag co-appearances can be used for sentiment overlap estimation. We calculated the relative co-occurrence frequencies of some hashtags and smileys. Table 5 shows some of the observed co-appearance ratios. As expected some of the observed tags frequently co-appear with other similar tags.

Hashtags	#happy	#sad	#crazy	# bored
#sad	12.8	-	-	-
#crazy	14.2	3.5	-	-
#bored	2.4	11.1	2.1	-
#fun	19.6	2.1	15	4.4
Smileys	:)	;)	:(	$X($
;)	35.9	-	-	-
:(	31.9	10.5	-	-
$X($	8.1	10.2	36	-
$:S$	10.5	12.6	21.6	6.1

Table 6: Percentage of shared features in feature vectors for different tags.

Interestingly, it appears that a relatively high ratio of co-appearance of tags is with opposite meanings (e.g., *"#ilove eating but #ihate feeling fat lol"* or *"happy days of training going to end in a few days #sad #happy"*). This is possibly due to frequently expressed contrast sentiment types in the same sentence – a fascinating phenomena reflecting the great complexity of the human emotional state (and expression).

5.3.2 Feature overlap

In our framework we have created a set of feature vectors for each of the Twitter sentiment tags. Comparison of shared features in feature vector sets allows us to estimate dependencies between different sentiment types even when direct tag co-occurrence data is very sparse. A feature is considered to be shared between two different sentiment labels if for both sentiment labels there is at least a single example in the training set which has a positive value of this feature. In order to automatically analyze such dependencies we calculate the percentage of shared Word/n-gram/Pattern features between different sentiment labels. Table 6 shows the observed feature overlap values for selected sentiment tags.

We observe the trend of results obtained by comparison of shared feature vectors is similar to those obtained by means of label co-occurrence, although the numbers of the shared features are higher. These results, demonstrating the pattern-based similarity of conflicting, sometimes contradicting, emotions are interesting from a psychological and cognitive perspective.

6 Conclusion

We presented a framework which allows an automatic identification and classification of various sentiment types in short text fragments which is based on Twitter data. Our framework is a supervised classification one which utilizes Twitter hashtags and smileys as training labels. The substantial coverage and size of the processed Twitter data allowed us to identify dozens of sentiment types without any labor-intensive manually labeled training sets or pre-provided sentiment-specific features or sentiment words.

We evaluated diverse feature types for sentiment extraction including punctuation, patterns, words and n-grams, confirming that each feature type contributes to the sentiment classification framework. We also proposed two different methods which allow an automatic identification of sentiment type overlap and inter-dependencies.

In the future these methods can be used for automated clustering of sentiment types and sentiment dependency rules. While hashtag labels are specific to Twitter data, the obtained feature vectors are not heavily Twitter-specific and in the future we would like to explore the applicability of Twitter data for sentiment multi-class identification and classification in other domains.

References

Akkaya, Cem, Janyce Wiebe, and Rada Mihalcea. 2009. Subjectivity word sense disambiguation. In *EMNLP*.

Andreevskaia, A. and S. Bergler. 2006. Mining wordnet for fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *EACL*.

Balog, Krisztian, Gilad Mishne, and Maarten de Rijke. 2006. Why are they excited? identifying and explaining spikes in blog mood levels. In *EACL*.

Bloom, Kenneth, Navendu Garg, and Shlomo Argamon. 2007. Extracting appraisal expressions. In *HLT/NAACL*.

Davidov, D. and A. Rappoport. 2006. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *COLING-ACL*.

Davidov, D. and A. Rappoport. 2008. Unsupervised discovery of generic relationships using pattern clusters and its evaluation by automatically generated sat analogy questions. In *ACL*.

Davidov, D., O. Tsur, and A. Rappoport. 2010. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *CoNLL*.

Esuli, Andrea and Fabrizio Sebastiani. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC*.

Jansen, B.J., M. Zhang, K. Sobel, and A. Chowdury. 2009. Twitter power: Tweets as electronic word of mouth. *Journal of the American Society for Information Science and Technology*.

Kim, S.M. and E. Hovy. 2004. Determining the sentiment of opinions. In *COLING*.

McDonald, Ryan, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. 2007. Structured models for fine-to-coarse sentiment analysis. In *ACL*.

Melville, Prem, Wojciech Gryc, and Richard D. Lawrence. 2009. Sentiment analysis of blogs by combining lexical knowledge with text classification. In *KDD*. ACM.

Mihalcea, Rada and Hugo Liu. 2006. A corpus-based approach to finding happiness. In *In AAAI 2006 Symposium on Computational Approaches to Analysing Weblogs*. AAAI Press.

Mishne, Gilad. 2005. Experiments with mood classification in blog posts. In *Proceedings of the 1st Workshop on Stylistic Analysis Of Text*.

Riloff, Ellen. 2003. Learning extraction patterns for subjective expressions. In *EMNLP*.

Strapparava, Carlo and Rada Mihalcea. 2008. Learning to identify emotions in text. In *SAC*.

Titov, Ivan and Ryan McDonald. 2008a. A joint model of text and aspect ratings for sentiment summarization. In *ACL/HLT*, June.

Titov, Ivan and Ryan McDonald. 2008b. Modeling online reviews with multi-grain topic models. In *WWW*, pages 111–120, New York, NY, USA. ACM.

Tsur, Oren, Dmitry Davidov, and Ari Rappoport. 2010. Icwsm – a great catchy name: Semi-supervised recognition of sarcastic sentences in product reviews. In *AAAI-ICWSM*.

Turney, Peter D. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *ACL '02*, volume 40.

Whitelaw, Casey, Navendu Garg, and Shlomo Argamon. 2005. Using appraisal groups for sentiment analysis. In *CIKM*.

Wiebe, Janyce and Rada Mihalcea. 2006. Word sense and subjectivity. In *COLING/ACL*, Sydney, AUS.

Wiebe, Janyce M. 2000. Learning subjective adjectives from corpora. In *AAAI*.

Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP*.

Wilson, Theresa, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399–433.

Yu, Hong and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *EMNLP*.