Target-Dependent Sentiment Analysis for Hashtags on Twitter

Zhiwei Yu[1] Raymond K. Wong[1] Chi-Hung Chi[2]

[1] University of New South Wales, Australia
{zhiweiyu,wong}@cse.unsw.edu.au
[2] CSIRO, Australia
chihung.chi@csiro.au

Technical Report
UNSW-CSE-TR-201329
October 2013

THE UNIVERSITY OF
NEW SOUTH WALES

School of Computer Science and Engineering
The University of New South Wales
Sydney 2052, Australia

Abstract

Microblogging services, such as Twitter, allow Internet users to exchange short messages easily. Users express their feelings on various topics as status messages or opinions. Hashtags are usually used in these services to mark essential words or phrases as a means for grouping topics. Thus, sentiment analysis on hashtags has become a popular method in determining user opinions on microblogs. In this paper, an effective approach to determine target-dependent hashtag sentiments is proposed. For a given tweet, a hashtag may carry different or even opposite opinions for different targets. Therefore, we aim to identify sentiment for two-dimensional parameters, namely $< hashtag, target >$. We firstly build a target-dependent tweet-level sentiment classifier based on target position sensitive features. A majority voting strategy for hashtag-level sentiment classification is then proposed as a baseline method. Finally, we show that its performance is significantly improved by propagation on a hyper relationship graph containing both target and hashtag nodes.

1 Introduction

Microblogging services, such as Twitter, allow users to exchange or broadcast small elements of content such as short sentences. Twitter is a very popular social network service where Internet users exchange status messages called tweets. Many of these tweets are related to certain popular or trendy topics such as the political events, hollywood news, electronic gadgets or consumer products.

According to the latest data on Wikipedia, Twitter has over 500 million users and the number of tweets published every day is over 340 million. As a result of this rapidly increasing number of tweets, the automatic classification of tweets according to their sentiment and topics has become a hot research topic. This is very useful to many applications such as marketing campaigns, market surveys and product feedback. For example, product vendors can perform analytics based on the sentiment classification to observe the public impression of their companies and products. They can also use this information to evaluate the effect of their advertising campaigns or propaganda events.

As mentioned in a few research papers (e.g., [4]), there has been a large amount of research already done in the area of sentiment analysis (SA) or opinion mining on tweets. Among these works, there are several kinds of definitions for sentiment, such as emotion sentiment [4] and target-dependent sentiment [6]. The aim of emotion sentiment is to detect the author's feelings in the texts, while target-dependent sentiment investigates the author's opinion about certain objects, such as products. Some times, these two types of sentiments may be different or even opposite. For example, in this tweet *"I am so sad to not have an iPhone"*, the author feels sad so that the emotion sentiment of this tweet is negative. On the other hand, since the author wants to obtain an iPhone, his opinion for the product *iPhone* is positive. In this case, target-dependent SA is more appropriate for the application of market survey or product feedback evaluation. Therefore, we mainly focus on target-dependent SA in this paper.

A hashtag is a word or phrase prefixed by a hash symbol, such as *#iphone*. It is designed to provide a way of grouping messages by searching for the hashtag to get a series of messages that contain it. It is also applied by Twitter users as a way to highlight topics or essential words in a message. Hashtags have been widely applied by users on Twitter. As introduced by [22], nearly 14.6% of tweets have at least one hashtags. If only subjective tweets (tweets with positive/negative sentiment expressions) are considered, this proportion increases to 27.5%. This result shows that hashtags have become an important metadata for researchers to interpret the enormous amount of information on Twitter.

Hashtags are also very useful for investigating the sentiment of tweets. Each hashtag corresponds to a series of tweets published by different users. Since one tweet may contain more than one hashtag and a single hashtag may appear in many different tweets, it provides an effective way to investigate the relationship between both hashtags and tweets. Further, if a hashtag with strong sentiments can be determined, we can make a very confident sentiment prediction to a larger series of tweets that contain it. For example, detecting that *#bigthumbproblems* and *#smallscreen* always occurrence together, the negative sentiment of *#bigthumbproblems* is shared to *#smallscreen* and further expanded to tweets containing *#smallscreen*.

The sentiments in hashtags are categorized into 2 types in [22].

- The first type is *sentiment hashtags*, which is composed of sentiment words only. Typical examples are #love and #sad, those hashtags contains sentiment words within it. However, for the application of product opinion, it is not always the case that sentiment words lead to hashtags with the same sentiments. For example, #sad always appears in some neutral tweets just talking about bad experiences such as a mobile phone is stolen or broken by some incident, but not that the product is bad itself. Another example is that #hate2wait, the sentiment word *hate* expresses negative feelings and is always included in negative lexicons. However, this hashtag are also used by users eager to get their favourite mobile phone. Further, many sentiment hashtags may not contain any sentiment words. For example, in the hashtag $\#shortbatterylife$, the author is implicitly not happy with his mobile phone. As a result, it is not always practical to detect target-dependent sentiment hashtags just by sentiment lexicons.

- The second type is *sentiment-topic hashtags* in which the topical word and the sentiment words appear together without separating blanks. A typical example is $\#loveiphone$, in which the word *love* is the sentiment word and *iphone* is the topic. At the same time, the topic word may not directly appear in the hashtag. For example, $\#smallscreen$ and $\#tinykeys$ can also be used to express negative sentiments to a target $iPhone5$.

Hashtags can also be target-dependent. The same hashtag contains different sentiments to different targets. Especially, in comparative texts, some hashtags always hold opposite sentiment with each other on a comparative target. Hence, detecting the relationships between different targets is significant for effective detection of the sentiment of these hashtags. For example, in the discussion about choosing mobile phones, the hashtag $\#teamiphone$ is always applied at the end of their tweets to express their opinions by authors who support iPhone and object to android. If we can determine that "*iphone*" and *"galaxy s4"* are comparative targets and the hashtag $\#teamiphone$ is positive to target "*iphone*", we can further predict that $\#teamiphone$ tends to have negative sentiment to target *"galaxy s4"*.

In this paper, a target-dependent hashtag sentiment classification approach is proposed. Detecting that same hashtags may have opposite opinions with different targets in the same tweet, our aim is to identify the sentiment for a two-dimensional parameters, namely $< hashtag, target >$. In this approach, we first build a target-dependent tweets-level sentiment classifier based on target position sensitive features. Later, a majority voting strategy for hashtag-level sentiment classification is proposed a baseline. Finally by performing a propagation on a Hyper Relationship Graph (HRG) that contains target and hashtags nodes, the performance of the hashtag-level classifier is significantly improved.

The organization of this paper is as follows. Section 2 summarizes the related work. Section 3 introduces the formal definition of the target-dependent hashtag sentiment classification problem. Section 4 discusses the techniques for building tweet-level target-dependent classifier. Section 5 presents the proposed Hyper Relationship Graph and hashtag-level sentiment classification approach. After that Section 6 presents experiment results for observing the dataset and studying the performance of classifiers. Finally, Section 7 concludes the paper.

2 Related Work

In the area of sentiment analysis, there has been a large amount of research. Sentiment analysis has be investigated on different levels. At first related research has mainly focused on classifying long texts, such as [15, 21, 28]. The task at this level is to detect whether a document expresses a wholely positive or negative sentiment. Instead of the SA research on document level, there is some work by researchers in the area of phrase level and sentence level sentiment classification recently [24, 25]. The task at this level goes to the sentences and determines whether each sentence expressed a positive, negative, or neutral opinion. Since both the document-level and sentence-level analyses do not discover what exactly people liked and did not like, The entity and aspect level sentiment analysis [5] is based on the idea that an opinion consists of both sentiment and target. The application of a sentiment without its target is limited.

Since words and phrases that convey positive or negative sentiments are instrumental to sentiment analysis, many researcher work on building sentiment lexicons, such as WordNet-based approaches [5, 23].

Topic model, such as LSA and LDA, are shown to be effective in classifying the short and sparse text by [17, 19]. Some works attempt to incorporate the sentiment factor into topic models to give the description about opinion generation [11, 13].

The sentiments are further classified into regular and comparative ones [8]. In this area, some researchers work on identifying the comparative sentence [7, 26]. Jindal et al. [7] classifies comparative sentiments into four types: non-equal gradable, equative, superlative, and non-gradable. They also showed that keywords and key phrases as features were already sufficient and SVM gave the best results on their dataset.

Recently, SA research has begun to pay more and more attention to social networks, such as Twitter, due to their growing user groups and significant influence on peoples' social lives. Tweets are very different from long, well composed texts. Tweets are more casual and limited to 140 characters of text. Further, Twitter users have different types of informal writing to express their thoughts, which brings extra challenges to traditional sentiment analysis approaches. Since there are a large range of topics discussed on Twitter, it is very difficult to manually collect enough data to train a sentiment classifier for tweets. However, noisy labels, such as :) and :(, are very effective to filter the training data as shown by [1, 4, 9, 18], which avoid the huge cost associated with hand-labelling. Davidov et al. [2, 3] and Jiang et al. [6] employ hashtags and smileys as sentiment labels for classification to allow diverse sentiment types for short texts.

Alec et al. provide a distant supervision approach to classifying the sentiment of twitter in [4]. They show that the different machine learning algorithms (Naive Bayes, Maximum Entropy and SVM) have accuracies of over 80% when trained with emoticon data. Pang et al. [17] investigated different features such as unigrams, bigrams, adjectives and POS-tags in their work. Their experimental results found that the SVM classifier with unigram presence features outperformed other competitors. Subhabrata et al. [14] consider 4 classes (positive, negative, objective and spam) in their approach. They obtain good performance on accuracy and precision on their manually annotated dataset. Barbosa and Feng [1] investigated a two-stage SVM (subjectivity and polarity)

classifier, which seems to be more robust regarding biased and noisy data. In this two-stage classifier, the first stage classifies the subjective (positive and negative) tweets from the objective (neutral) ones. Davidov et al. [20, 3] follow a classic semi-supervised learning framework for extending the training dataset.

While many state-of-the-art approaches for SA adopt the target-independent strategy, which may assign irrelevant sentiments to the given target, Jiang et al. [6] focus on target-dependent tweet-level sentiment classification. That is, given a query, the sentiments of the tweets are classified as positive, negative or neutral depending on whether they contain positive, negative or neutral sentiments on that query. In their approach, the query serves as the target of the sentiments.

Wang et al. [22] observed the hashtag-level sentiment classification problem. They propose a graph-based hashtag sentiment classification approach, which follows the 2 stage approach described in [1]. They discuss several graph-based approaches based on hashtag-hashtag relationships. This work is the most related to this paper. However, their approach is based on target-independent features and only involves hashtag-hashtag relationship in their hashtag relationship graph. In contrast, in this paper, we show that target-dependent hashtag sentiment classification is, though it is more difficult problem, more practical for the applications such as market survey and product reviews/evaluations; and target-target relationships are essential to build the target-dependent hashtag sentiment classifier. To the best of our knowledge, the approach proposed in our paper is the first one focusing on target-dependent hashtag sentiment classification; and involving target-hashtag and target-target relationships to further tune the classifier's performance.

The contributions of this paper are summarized as follows:

- For classifying comparative sentiments, position-sensitive features are selected to capture different features for different targets within the same tweet text. A mixed training data set with both multi-target and single-target tweets are then used for fitting both regular and comparative sentiments.

- A HRG graph with target nodes and hashtag nodes is applied to tune the sentiment classifier for hashtags. We observe the approaches to measure the relationship between targets and hashtags and whether a hashtag is appropriate to be involved in a sentiment propagation.

- We propose a target-dependent loop belief propagation algorithm for transferring sentiment on the HRG, which significantly improves the performance of the hashtag sentiment classifier compared with a majority-vote based line.

3 Problem Definition

In this section, the formal definition for the task of target-dependent hashtag sentiment classification is first proposed. Given a set of hashtags $H = \{h_1, h_2, ..., h_m\}$, where each hashtag h_i is associated with a set of tweets $TW = \{tw_1, tw_2, ..., tw_n\}$ and a set of targets $T = \{t_1, t_2, ..., t_k\}$. We aim to collectively infer the sentiment polarities, $y = \{y_1, y_2, ..., y_{m*k}\}$ where $y_i \in S = \{pos, neg, neu\}$, for set $\{< h, t > | h \in H, t \in T\}$.

For the convenience of presentation, necessary formulations for expressing the tweet set and relationships are introduced in table 3.1. In these formulations, we apply notation $h \in tw$ to express the tweet tw containing hashtag h in its text and $t \in tw$ to express the tweet tw containing target t in its text.

Table 3.1: Formulations

Notation	Definition
label(tw,t)	The sentiment label of tweet tw to target t
TW(h)	$\{tw \mid tw \in TW \wedge h \in H \wedge h \in tw\}$
TW(t)	$\{tw \mid tw \in TW \wedge t \in T \wedge t \in tw\}$
TW(h,t)	$\{tw \mid tw \in TW \wedge h \in H \wedge t \in T \wedge h \in tw \wedge t \in tw\}$
TW(h,t,s)	$\{tw \mid tw \in TW \wedge h \in H \wedge t \in T \wedge s \in S \wedge h \in tw \wedge t \in tw \ \wedge label(tw,t) = s\}$
MT(h)	$\{tw \mid tw \in TW \wedge h \in H \wedge h \in tw \wedge \exists t \in tw, t' \in tw \wedge t \neq t'\}$
TS(t,t')	$\{tw \mid tw \in TW \wedge t \in T \wedge t' \in T \wedge t \in tw \wedge t \in tw \wedge t \neq t' \wedge label(tw,t) = label(tw,t')\}$
TD(t,t')	$\{tw \mid tw \in TW \wedge t \in T \wedge t' \in T \wedge t \in tw \wedge t \in tw \wedge t \neq t' \wedge label(tw,t) \neq label(tw,t')\}$

The hashtag-level sentiment classification is inherently highly related to the tweet-level sentiment analysis results. In a target-dependent tweet-level classifier, for each target t, each tweet tw can be assigned a positive, negative or neutral probability $P(tw,t,pos)$, $P(tw,t,neg)$ and $P(tw,t,neu)$. It needs to ensure that $P(tw,t,pos) + P(tw,t,neg) + P(tw,t,neu) = 1$. Correspondingly, in a target-dependent hashtag sentiment classifier, for each target t, each hashtag h can be assigned a positive, negative or neutral probability $P(h,t,pos)$, $P(h,t,neg)$ and $P(h,t,neu)$, also ensuring that $P(h,t,pos) + P(h,t,neg) + P(h,t,neu) = 1$.

Mainly inspired by Jiang et al. [6] and Wang et al. [22], we design a three-step approach in this paper:

- Target-dependent tweet-level sentiment classification. It is further divided into two stages: subjectivity classification and polarity classification. The first stage subjectivity classification is for deciding if the tweet is subjective or neutral about a target. The second stage polarity classification is for deciding if subjective tweets are positive or negative.

- Basic sentiment probability evaluation for hashtags. This is done by aggregating the sentiment of all tweets related to a single hashtag to generate the sentiment probability.

- Performance tuning based on the HRG graph. Detecting the relationships between different hashtags and targets, we introduce a HRG graph to capture all the relationships. By applying a loop belief propagation algorithm on this graph, we iteratively update the probability of each $< hashtag, target >$ pair.

4 Tweet-Level Comparative Sentiment Classifier

We adopted the state-of-the-art tweet-level sentiment classification approach from [1, 22], which applies a two-stage SVM classifier to determine the sentiment polarity of a tweet.

- The first stage subjectivity classifier determines whether a tweet is neutral or subjective;
- The second stage polarity classifier assigns a subjective tweet with positive or negative polarity.

The Scikit-Learn SVM package [16] is used in our experiments for building the classifier. In this section, we discuss building of the training dataset and the target-dependent feature selection.

4.1 Mixed Training Dataset

The cost for building a practical, manually annotated training dataset is always huge. Current state-of-the-art approaches, such as [4], tend to use special emoticons or hashtags with sentiments to filter enough training samples. For tweets with only a single target, it is easier to collect training samples because, in most cases, the emotions of the author is strong relative to his sentiment to the target. For example,"my iphone charger is broken again #disappointed". The author is not satisfied with his iphone charger, which can also be detected by the hashtag *#disappointed* in this case.

However, for tweets with more than one target, it is hard to automatically collecting training samples. For example, "finally get my new iphone5, bye samsung! #nice". There are two targets in this tweets, and we can automatically detect that the author is pretty happy based on the hashtag #nice. However, we can not know the author is happy with which target based on this hashtag.

There are some hashtags themselves including a target. For example, *#teamiphone* or *#byeapple*. However, training samples collected by these very limited hashtags are not general enough to capture all the features. Further, there are also neutral training samples, which make it hard to find appropriate hashtags to collect.

In order to capture features for both tweets with single targets and multiple targets, the training data set is built from the following two resources:

- Tweet samples with only one target collected by filtering rules based on a sentiment lexicon. All these training samples contains some hashtags - part of which exist in the sentiment lexicon from [12]. As mentioned previously in section 1, according to the special language customs on Twitter, we remove some unreliable words from this lexicon, such as "*addicted*" and "*sad*". The hashtags used for building this dataset are removed from the tweets to prevent them from having too much weight in the classifier.
- Tweet samples which contain more than one target annotated by human annotators. A very high proportion of these samples are tweets with the topic of comparison between two or more targets.

4.2 Target-Dependent Feature Selection

Most traditional features for text classification cannot be directly applied to capture target-dependent features. The primary reason is that there may be more than one target holding different sentiments in the same tweet. For example:

"I think iphone is better than galaxy too"

There are two targets in this tweet: *iphone* and *samsung galaxy*. This tweet contains positive sentiments for the target *iphone*, but negative sentiments for *samsung galaxy*. Since both targets appear in the same text, if we simply apply TF-IDF features for this sample, the features for sentiments for both targets will be the same. It is necessary to attach more information to distinguish the features for different targets in the same text.

Jiang et al. [6] proposed some rules based on a syntactic parse tree, to attach both syntactic information and position information to words. Their approaches works well for both subjectivity classification and polarity classification on their data set. Different from this approach, we do not rely on a syntactic parse tree and several feature selection rules to build target-dependent features. Our approach for building the target-dependent features by attaching the relative position against the target to the words, is simpler yet more general.

We first detect the positions of all the targets in a tweet. Then, for each target, we generate a different feature list in which each feature is a word appended by a postfix of the relative position against this target. There are three kinds of relative positions as show in table 4.1.

Table 4.1: Relative Position Definition

Position	Meaning	Postfix
Left	Current word is on the left side of target, but not on the left side of any other target	_l
Right	Current word is on the right side of target, but not on the right side of any other target	_r
Far	Current word is in the same direction of target and any other target	_f

For the tweet example in the beginning of this subsection, the extracted target-dependent features are shown in table 4.2.

After extracting features, one tweet becomes several feature vectors, each of which correspond to a different target. Further, we calculate TF-IDF value for these new features to change it into a numerical vector.

5 Target-dependent Hashtag Sentiment Classifier

In this section, we discuss the target-dependent hashtag sentiment classification approach based on the tweet-level sentiment classifier described in section 4.

Generally, although some hashtags contain strong sentiments in most cases, there are still a few tweets with these hashtags that do not contain sentiments. Further, the tweet-level sentiment classifier may not be accurate enough. As a result, for a hashtag and target, the related tweets set contains tweets with all kinds of sentiments.

5.1 The Majority Voting Classifier Baseline

Majority voting strategy is an intuitive method to aggregate the labels of small classes in order to generate the label of their super class. The main idea of this strategy is to choose the label with the largest probability. It is also widely applied in the ensemble learning area to determine the final output by multiple independent classifiers [10]. We describe this strategy in the format of hashtag sentiment classification as follows:

$$P(h,t,s) = \frac{|TW(h,t,s)|}{|TW(h,t)|} \tag{5.1}$$

$$S(h,t) = \underset{s \in S}{argmax} P(h,t,s) \tag{5.2}$$

As shown in section 6, this strategy works well for polarity classification, but its recall performance for subjectivity classification is not promising. This is mainly because tweet-level classifiers are not accurate enough. This is our main motivation to further explore more advanced strategies, so that we can improve the performance. One direction for this task is to first improve the performance of the tweet-level classifier. An alternative way for tuning the performance is to utilize the sufficient relationship information on hashtag and target level.

5.2 The Hyper Relationship Graph

The idea of using HRG is motivated by observing that there are relationships between different targets. These target relationships can be treated as one of 2 types:

- Same-group targets: such as *apple* and *iphone*, *galaxy* and *android*.

Table 4.2: Target-Dependent Feature Extraction Examples

words	iphone		galaxy	
	relative position	feature	relative position	feature
I	left	i_l	far	i_f
think	left	think_l	far	think_f
is	right	is_r	left	is_l
better	right	better_r	left	better_l
than	right	than_r	left	than_l
too	far	too_f	right	too_r

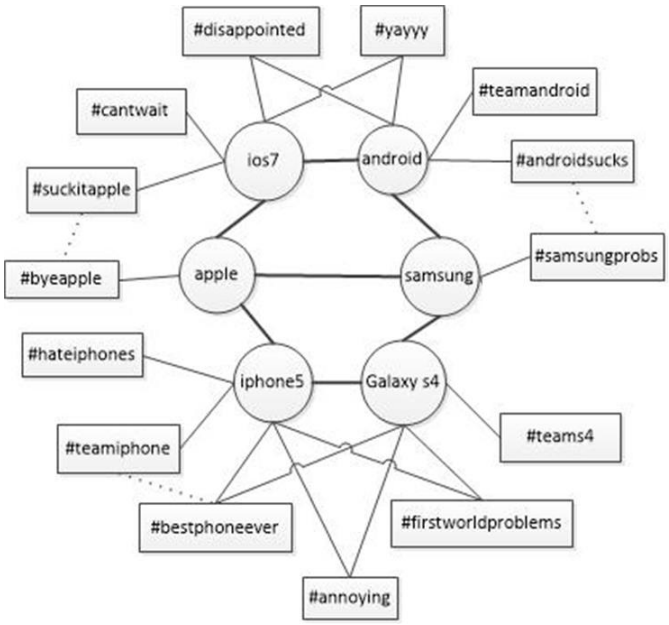


Figure 5.1: The Hyper Relationship Graph

- Comparative targets: such as *iphone*5 and *samsung galaxy s4*, *apple* and *samsung*.

Same-group targets tend to share the same sentiments on all hashtags, while comparative targets tend to share opposite sentiments on certain hashtags. For example, suppose we know that *#teamiphone* is a positive hashtag for target *iphone*, it also tends to have negative sentiment to *samsung galaxy*.

To capture these relationships, we propose a HRG that contains two kinds of nodes: target nodes and hashtag nodes. Respectively, there are 3 kinds of edges in this graph: target-target, hashtag-hashtag and target-hashtag. Figure 5.1 illustrates a typical example of HRG.

- The circles in this graph illustrate target nodes, which are connected with each other by target-target relationship edges. For example, the target node *iphone*5 has both connections to target node *apple* and target node *Galaxy S4*. However, target node *apple* has a positive influence on target node *iphone*5, while target node *GalaxyS*4 has a negative influence on it.

- The rectangles in this graph illustrate hashtag nodes, which have connections with target nodes and other hashtag nodes. For example, the hashtag node *#byeapple* has a negative sentiment to target node *apple*. The same sentiment from this hashtag can be propagated to the target node *iphone*5, which has a positive relationship with target node *apple*, while the opposite sentiment from this hashtag can be propagated to the target node *android* through the negative relationships between target node *apple* and *samsung* and positive relationships between the target node *samsung* and *android*.

- There are also relationships between hashtags. For example, *#teamiphone* and *#bestphoneever* has a high co-occurrence probability on target *iphone*5. They tend to hold the same sentiment with this target.

5.3 Quantification of Relationships on HRG

To quantify the relationship between two targets, we introduce the function RT(t, t').

$$RT(t,t') = log(|TW(t,t')|) * \frac{|TS(t,t')| - |TD(t,t')|}{|TW(t,t')|} \tag{5.3}$$

TS(t,t') is the tweets set in which target t and t' hold the sample sentiment, while TD(t,t') is the tweets set in which target t and t' hold the opposite sentiment. TW(t,t') is the tweets set that contains both t and t'.

The value of RT(t, t') value is directly proportional to the probability that target t and t' share the same sentiment in a tweet. If the —TS(t,t')— < —TD(t,t')—, RT(t, t') will return subtractive value. Since the statistic results with more tweets are more reliable, the weight log(—TW(t,t')—) is applied to function RT(t,t') to increase the influence of target-target relationships, which has more co-occurrences.

RT(T,T') is designed to find comparative targets. However, not all subjective hashtags always hold opposite sentiments with comparative targets. Instead, there are two kinds of subjective hashtags:

- The hashtags with target-independent sentiments, such as *#disappointed* and *#bestphoneever*,which can be applied to express the same sentiment to comparative targets.
- The hashtags with target-dependent sentiments, such as *#byeapple* and *#teamiphone*, which can only contain different sentiments with comparative targets.

Only target-dependent sentiment hashtags can be applied to generate the negative sentiment influence between comparative targets. To distinguish between these 2 kinds of hashtags, we introduce the function HT(h) for quantifying if the hashtag is frequently used in comparative sentiment.

$$HT(h) = \frac{|MT(h)|}{|TW(h)|} \tag{5.4}$$

MT(h) is the tweets which contain h and at least 2 different targets, so that HT(h) is the probability that hashtag h appears in a tweet with at least 2 different targets. Though not all tweets with 2 or more targets are comparative, all tweets with only 1 target should not be comparative. Thus, target-independent sentiment hashtags will have low HT(h) value.

As discussed in [22], hashtags co-occurring in tweets have a much higher probability to share the same sentiment polarity than if they are randomly selected. For quantifying the relationship between two hashtags, the function RH(h, h', t) is applied.

$$RH(h,h',t) = \frac{|TW(h,t) \cap TW(h',t)|}{|TW(h,t) \cup TW(h',t)|} \tag{5.5}$$

To avoid assigning subjective hashtags to irrelevant targets, we extend the normal co-occurrence function by involving the target. For example, *#bigscreen* and *#thebest* are hashtags with high co-occurrences in tweets talking about the target *samsung*, so transferring positive sentiment from *#thebest* to *#bigscreen* for target *iphone*5 is not appropriate.

5.4 The Loop Belief Propagation(LBP) on Hyper Graph

The main idea of LBP is to classify each node in a graph through belief message passing in an iterative manner. As discovered by [27] and [22], LBP shows good performance after any number of iterations in practice, although not guaranting convergence.

The propagation function is defined as follows:

$$P_{i+1}(h,t,s) = \frac{P_i(h,t,s) + \alpha * PH_i(h,t,s) + \beta * PT_i(h,t,s)}{\theta} \tag{5.6}$$

In each iteration, each probability is modified by involving the hashtag-hashtag influence $PH_i(h,t,s)$ and target-target influence $PT_i(h,t,s)$. α and β are constant weight, while θ is a provisional computed normalized factor, where $P_{i+1}(h,t,pos) + P_{i+1}(h,t,neg) + P_{i+1}(h,t,neu) = 1$.

The hashtag-hashtag influence function $PH_i(h,t,s)$ is defined as a probability summation of all the related hashtags weighted by hashtag-hashtag relationship RH(h, h', t).

$$PH_i(h,t,s) = \sum_{h' \in H} RH(h,h',t) * (P_i(h',t,s) - \delta) \tag{5.7}$$

δ is the constant threshold between 0-0.5 for controlling the influence of P(h', t, s). Basically, if $P_i(h',t,s) < \delta$, the $PH_i(h,t,s)$ tends to reduce the value of $P_{i+1}(h,t,s)$, otherwise it tends to increase the value of $P_{i+1}(h,t,s)$.

The target-target influence function $PT_i(h,t,s)$ is defined as a probability summation of all the related targets weighted by target-target relationship $\gamma * |HT(h) * RT(t,t')|$.

$$PT_i(h,t,s) = \sum_{t' \in T} \gamma * |HT(h) * RT(t,t')| * (P_i(h,t',s) - \delta) \tag{5.8}$$

$$\gamma = \begin{cases} -1 & \text{if HT(t)¿0.2} \wedge \text{RT(t, t')} \text{ ¡ } 0 \text{ ,} \\ 1 & \text{otherwise.} \end{cases} \tag{5.9}$$

γ is the key factor to control if the sentiment influence is positive or negative. The target influence can only be subtractive when HT(h) is big enough and RT(t, t') is negative, which means the hashtag h is a comparative hashtag and target t and t' are comparative targets. The threshold value 0.2 is carefully chosen by observing the distribution of HT(h), which is shown in the section 6.

The whole algorithm is summarized in algorithm 1.

- Firstly, all values for RH(h,h',t), RT(t,t') and HT(h) are calculated.
- Next, the propagation loop begins. In each round of the loop, all P(h, t, s) values are recalculated according to the propagation function.

- The mean absolute error between adjacent rounds are calculated to determine if the propagation loop should stop.
- Finally, output the s labels with the highest $P(h,t,s)$ value as the sentiment output for each pair $< h,t >$.

The α, β and δ are the constant parameters in this algorithm. After applying the grid search method to find the parameters which provide the best accuracy performance, α is set to 0.02, β is set to 0.02 and δ is set to 0.3.

We use a pair $< \#teamsamsung, iphone >$ as an example to demonstrate how the target-target relationship works in this algorithm.

The original input for a majority baseline for this case is as follows:

- $P(\#teamsamsung, iphone, pos)$= 0.26
- $P(\#teamsamsung, iphone, neg)$ = 0.28
- $P(\#teamsamsung, iphone, neu)$=0.46

As a result, the voting baseline will consider the sentiment as neutral.

In the HRG propagation process, the PT influence for $P_0(\#teamsamsung, iphone, pos)$ in the first round of the sentiment propagation loop is shown in table 5.1.

Table 5.1: PT sentiment influence for $P(\#teamsamsung, iphone, pos)$

Target t'	TR(*iphone*,t')	P(*#teamsamsung*, t',*pos*)-δ	Influence
htc	-4.3346072	-0.133333	0.93455054
samsung	-6.7256516	0.018966	-0.20626498
s4	-1.8569533	0.06	-0.18016384
galaxy	-4.842784	0.030508	-0.23890447
android	-5.9220900	0.02	-0.1915227
aple	5.1181873	-0.054386	-0.27835773
iphone5	1.0980763	-0.188889	-0.20741454

- The calculated $HT(\#teamsamsung)$ value is 0.36. Since $HT(\#teamsamsung)$ <0.2, the comparative target *htc*, *samsung*, *s*4, *galaxy* and *android* can be applied to propagate the sentiment in the reverse direction according to equation 5.9.
- After this round of propagation, the $PT_0(\#teamsamsung, iphone, pos)$=-0.007361555, which causes $P_1(\#teamsamsung, iphone, pos) < P_0(\#teamsamsung, iphone, pos)$. So the sentiment for $< \#teamsamsung, iphone >$ tends to be negative.

6 Experiments

6.1 Data Collection and Evaluation

In our experiment, we main focussed on tweets relating to mobile phone discussions. We first manually collected a target list of 45 items, such as *iphone*,

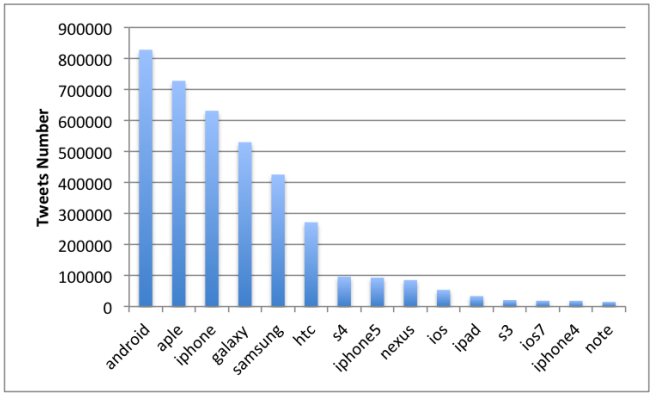


Figure 6.1: The Number for Tweets Related to Different Targets

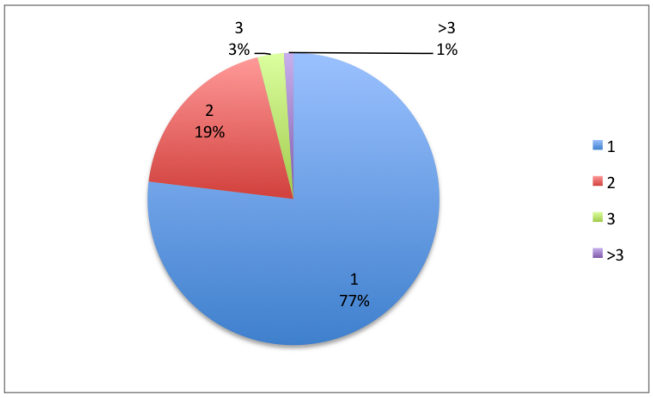


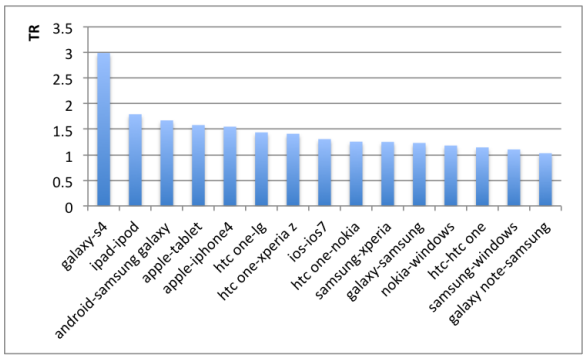
Figure 6.2: The Proportion of Tweets with Different Number of Targets

android, *samsung*, *nexus*, *htc one*. After that, we randomly crawled for tweets on the Twitter web site to collect tweets text that contains at least one of the targets. We also removed retweets and only kept English tweets. As a result, we obtained 2,012,451 English original tweets from 1,032,505 unique twitter users. The top 15 tweets related to targets are shown in figure 6.1.

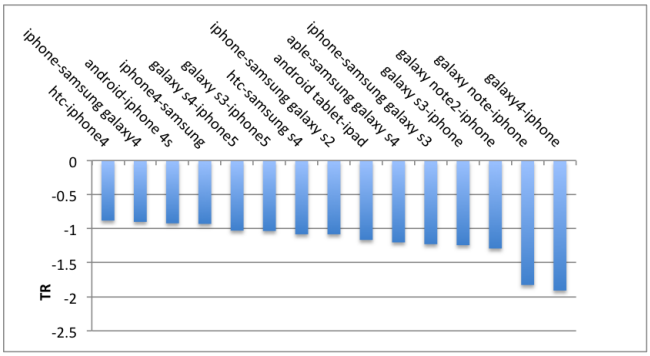
In figure 6.2, the proportion of tweets with a different number of targets are shown. The number of tweets with only one target has the highest proportion of 77%. The number of tweets with 2 targets has the second highest proportion of 19%. Though much less than single-target tweets, the tweets with 2 or more targets still hold a significant proportion among all tweets. More importantly, they provide more sufficient information for comparative sentiment for the application of market surveys or product feedback evaluations.

In figure 6.3(a) and 6.3(b), the top same-group and comparative target pairs are shown. In these two figures, the x-axis is the target pair and the y-axis is the TR function value. From these 2 figures, comparative target pairs, such as *galaxy*4 and *iphone*, tend to hold lower TR value, while the same-group target pairs, such as *galaxy* and $s4$, tend to hold higher TR value.

In figure 6.4, the HT function value of essential hashtags are shown. Al-



(a) The Highest TR Values (Top Same Group Targets)



(b) The Lowest TR Value (Top Comparative Targets)

Figure 6.3: The TR Value for Measuring Target-Target Relationships

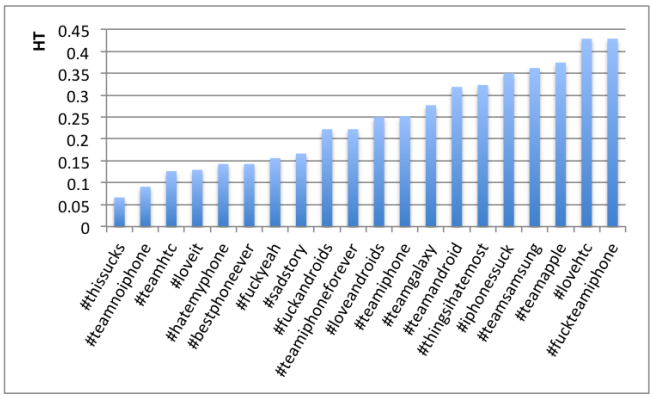


Figure 6.4: The HT Value of Hashtags

though there are a few bad cases, such as *#teamhtc*, which has a low HT value, and *#thingsihatemost*, which has a high HT value, most of the hashtags with target-independent sentiments, such as *#thissucks* and *#bestphoneever* tend to have a low HT value and most of the hashtags with target-dependent sentiment, such as *#teamsamsung* and *#lovehtc*, tend to have a high HT value.

6.2 Performance Study

We conducted several experiments to evaluate subjectivity classifiers. The mixed training dataset for training the target-dependent tweet-level sentiment classifier is built from two sources.

- A human-annotated training dataset which contains 1197 tweets, where each tweet contains more than one target. For each target, the annotator assigns a *positive*, *negative* or *neutral* mark for it.
- An automatically generated training dataset which contains 4000 tweets, where each tweet contains one target. These tweets all contain at least one hashtag, part of which matches sentiment lexicon from [12]. Each of these samples only has either *positive* or *negative* labels.

For training the subjectivity classifier, a total of 948 subjective and 1455 neutral $< tweet, target >$ pairs are applied as the training dataset. For training the polarity classifier, a total of 3077 positive and 2916 negative $< tweet, target >$ pairs are applied as the training dataset.

Our performance evaluation criteria consist of *accuracy*, *precision*, *recall* and *F1-score*.

We build a two-stage SVM tweet-level classifier with 10-fold cross validation. 200 $< target, tweet >$ pairs are randomly selected and manually annotated as positive, negative and neutral for evaluating the recall performance of the subjectivity classifier. The precision performance of the subjectivity classifier is evaluated on the dataset contains 200 $< target, tweet >$ pairs from the output of the subjectivity classifier. The precision of the tweet-level polarity classifier is evaluated on a dataset of 100 $< target, tweet >$ pairs for either of positive and negative classes randomly selected from its output. We also randomly select 200

$< target, tweet >$ subjective pairs from the output of the subjectivity classifier and manually annotate them as positive, negative and neutral for evaluating the recall performance of the polarity classifier for both positive and negative classes. The performance of the two stages of the tweet-level sentiment classifier is shown in table 6.1.

Compared to [6], which gains maximum 68.2% accuracy for subjectivity classifier and maximum 85.6% accuracy for polarity classifier, the performance of our target-dependent tweet-level sentiment classifier is satisfactory. However, since we mainly focus on hashtag-level sentiment classification, tuning our approach further to improve this performance is outside the scope of this paper.

Table 6.1: Performance for Tweet-Level Sentiment Classifier

Classifier	Subjectivity	Polarity
Accuracy	74.23%	75.04%
Precision	65%	Pos: 79.2% Neg: 70.6%
Recall	63%	Pos: 74.2% Neg:76.2%
F-score	64%	Pos: 76.6% Neg:73.3%

Table 6.2: Performance for Hashtag Sentiment Classifier

Classifier	Majority Voting Baseline	HRG Tuning
Precision	61.2%	66%
Recall	32%	64.5%
F1-score	42%	65.2%

Since the tweet-level sentiment performance is not very high, there is a high risk for a naive majority voting strategy for the hashtags sentiment classifier.

Since the majority voting baseline is based on statistics, it will only work well when —TW(h,t)— is big enough. Hence we set this threshold to be —TW(h,t)—¿10. After filtering by this threshold, there are total 8233 $< hashtag, target >$ pairs left; and are used as the candidates for the hashtag sentiment classification. 200 $< hashtag, target >$ pairs are then randomly selected from the output of the hashtag sentiment classification and are later manually marked with positive, negative and neutral labels to evaluate the recall performance. 100 $< hashtag, target >$ pairs of the either positive or negative classes are randomly selected from the output of Majority Voting Baseline and HRG Tuning. They are manually verified and marked with positive, negative and neutral labels for evaluating the precision performance (note that some positive or negative cases will be marked as neutral when they are found to be neither positive nor negative by our manual judgment).

The performance of the hashtag-level sentiment classifier is shown in table 6.2. The HRG Tuning performs better than the voting baseline on both *precision* and *recall*. In particular, the HRG Tuning is significantly better than the voting baseline in the recall performance.

7 Conclusions

In this paper, a target-dependent sentiment classification approach for twitter hashtags is proposed. For market surveys and evaluation of product feedback, our aim is to determine the sentiment label for a 2-dimension pair $< hashtag, target >$.

We first built a target-dependent tweet-level sentiment classifier. We built a training dataset mixed with an automatic generated tweets dataset with only one target, and human annotated tweets data, aiming to capture the feature of comparative sentiment features. We also proposed a general target-dependent features generation method, which works well for capturing different features for different targets that appear in the same tweet. After that, we trained a classifier and gained a satisfactory performance with 74.23% accuracy for subjectivity classifier and 75.04% accuracy for polarity classifier. (For example, in paper [6], the best accuracy for subjectivity classifier is 68.2% and the best accuracy for polarity classifier is 85.6%).

For the hashtag sentiment classifier, we initially introduced a majority voting hashtag-level sentiment baseline, which gained 61.2% in precision and only 32% in recall. Afterwards, the performance of the majority voting baseline was enhanced by LBP strategy on a hyper relationship graph. Experiment showed that the recall performance of the subjective classifier was significantly improved to 64.5% using the propagation algorithm.

Our future work is to improve the subjectivity classifier for both tweet-level and hashtag-level sentiment. The output of the hashtag sentiment classifier can also be applied to further tune the performance of the tweet-level sentiment classifier. These types of iterations can build a more accurate classifier.

Bibliography

[1] L. Barbosa and J. Feng. Robust sentiment detection on twitter from biased and noisy data. In *Computational Linguistics Posters*, pages 36–44, 2010.

[2] D. Davidov, O. Tsur, and A. Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *23rd International Conference on Computational Linguistics: Posters (COLING)*, pages 241–249, 2010.

[3] D. Davidov, O. Tsur, and A. Rappoport. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Fourteenth Conference on Computational Natural Language Learning (CoNLL)*, pages 107–116, 2010.

[4] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. Technical report, Stanford University, 2009.

[5] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004.

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

[7] Nitin Jindal and Bing Liu. Identifying comparative sentences in text documents. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 244–251. ACM, 2006.

[8] Nitin Jindal and Bing Liu. Mining comparative sentences and relations. In *AAAI*, volume 22, pages 1331–1336, 2006.

[9] A. Joshi, A. R. Balamurali, P. Bhattacharyya, and R. Mohanty. C-feel-it: a sentiment analyzer for microblogs. In *ACL Demo Papers (HLT)*, 2011.

[10] Sotiris B Kotsiantis, ID Zaharakis, and PE Pintelas. Supervised machine learning: A review of classification techniques, 2007.

[11] C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *18th ACM conference on Information and knowledge management (CIKM)*, pages 375–384, 2009.

[12] Bing Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, 2012.

[13] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *16th international conference on World Wide Web (WWW)*, pages 171–180, 2007.

[14] Subhabrata Mukherjee, Akshat Malu, Balamurali A.r., and Pushpak Bhattacharyya. Twisent: A multistage system for analyzing sentiment in twitter. In *21st ACM international conference on Information and knowledge management (CIKM)*, pages 2531–2534, 2012.

[15] Bo Pang, Lillian Lee, and Shivakumar 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.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[17] Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *17th international conference on World Wide Web*, 2008.

[18] J. Read. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *43nd Meeting of the Association for Computational Linguistics. Association for Computational Linguistics (ACL)*, 2005.

[19] Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. Short text classification in twitter to improve information filtering. In *33rd international ACM SIGIR conference on Research and development in information retrieval*, 2010.

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

[21] Peter D Turney and Michael L Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346, 2003.

[22] Xiaolong Wang, Furu Wei, Xiaohua Liu, Ming Zhou, and Ming Zhang. Topic sentiment analysis in twitter: A graph-based hashtag sentiment classification approach. In *20th ACM international conference on Information and knowledge management (CIKM)*, pages 1031–1040, 2011.

[23] Gbolahan K Williams and Sarabjot Singh Anand. Predicting the polarity strength of adjectives using wordnet. In *ICWSM*, 2009.

[24] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, 2005.

[25] T. Wilson, J. Wiebe, and P. Hoffmann. Recognizing contextual polarity: an exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399–433, 2009.

[26] Seon Yang and Youngjoong Ko. Extracting comparative entities and predicates from texts using comparative type classification. In *ACL*, pages 1636–1644, 2011.

[27] Jonathan S Yedidia, William T Freeman, Yair Weiss, et al. Generalized belief propagation. In *NIPS*, volume 13, pages 689–695, 2000.

[28] L. Zhuang, F. Jing, and X. Y. Zhu. Movie review mining and summarization. In *15th ACM international conference on Information and knowledge management (CIKM)*, pages 43–50, 2006.

Algorithm 1 Loop Belief Prorogation on HRG

$S \leftarrow \{pos, neg, neu\}$
for all $h \in H, h' \in H$ **do**
  if $h \neq h'$ **then**
    for all $t \in T$ **do**
      calculate(RH(h,h',t))
    end for
  end if
end for
for all $t \in T, t' \in T$ **do**
  if $t \neq t'$ **then**
    calculate(RT(t,t'))
  end if
end for
for all $h \in H$ **do**
  calculate(HT(h))
end for
$error \leftarrow 1$
$i \leftarrow 0$
while $error > 0.001$ **do**
  $count \leftarrow 0$
  for all $h \in H$ **do**
    for all $t \in T$ **do**
      for all $s \in S$ **do**
        $PH_i(h,t,s) \leftarrow \sum_{h' \in H} RH(h,h',t) * (P_i(h',t,s) - \delta)$
        $PT_i(h,t,s) \leftarrow \sum_{t' \in T} \gamma * |HT(h) * RT(t,t')| * (P_i(h,t',s) - \delta)$
        $P_{i+1}(h,t,s) \leftarrow (P_i(h,t,s) + \alpha * PH_i(h,t,s) + \beta * PT_i(h,t,s))/\theta$
        $error \leftarrow |P_{i+1}(h,t,s) - P_i(h,t,s)|$
        $count \leftarrow count + 1$
      end for
    end for
  end for
  $error \leftarrow error/count$
  $i \leftarrow i + 1$
end while
for all $h \in H$ **do**
  for all $t \in T$ **do**
    $y(h,t) \leftarrow argmax_{y \in S} P_{i+1}(h,t,s)$
  end for
end for