Normalization of Term Weighting Scheme for Sentiment Analysis

Alexander Pak[1], Patrick Paroubek[1(✉)], Amel Fraisse[1], and Gil Francopoulo[2]

[1] LIMSI-CNRS, Université Paris-Sud, 133, 91403 Orsay Cedex, France
irokez@gmail.com, {pap,fraisse}@limsi.fr
[2] TAGMATICA, 126 rue de Picpus, 75012 Paris, France
gil.francopoulo@tagmatica.com

Abstract. N-gram models with a binary (or tf-idf) weighting scheme and SVM classifiers are commonly used together as a baseline approach in lots of research studies on sentiment analysis and opinion mining. Other advanced methods are used on top of this model to improve the classification accuracy, such as generation of additional features or using supplementary linguistic resources. In this paper, we show how a simple technique can improve both the overall classification accuracy and the classification of minor reviews by normalizing the terms weights in the basic bag-of-words method. Any other term selection scheme may also benefit from this improved weighting scheme, if it is based on the n-gram model. We have tested our approach on the movie review and the product review datasets in English and show that our normalization technique enhances the classification accuracy of the traditional weighting schemes. The question whether we would observe similar performance increases for other language families is still to be investigated, but our weighting scheme can easily address any other language, since it does not use any language specific resource apart from a training corpus.

1 Introduction

The increase of the interest in sentiment analysis is usually associated with the appearance of web-blogs and social networks, where users post and share information about their likes/dislikes, preferences, and lifestyle. Many websites provide an opportunity for users to leave their opinion on a given object or a topic. For example, the users of IMDb[1] website can write a review on a movie they have watched and rate it on 5-star scale. As a result, given a large number of reviews and rating scores, the IMDb reflects general opinions of Internet users on movies. Many other movie-related resources, such as cinema schedule websites, use the information from the IMDb to provide information about the movies including the average rating. Thus, the users who write reviews on IMDb

After his PhD. at LIMSI-CNRS in 2012, Alexander Pak has joined Google in Zürich (alexpak@google.com)

[1] The Internet Movie Database: http://imdb.com.

Z. Vetulani and J. Mariani (Eds.): LTC 2011, LNAI 8387, pp. 116–128, 2014.
DOI: 10.1007/978-3-319-08958-4_10

influence the choice of other users, who will have a tendency to select movies with higher ratings.

Another example is social networks. It is popular among users of Twitter[2] or Facebook[3] to post messages that are visible to their friends, with an opinion on different consumer goods, such as electronic products and gadgets. The companies who produce or sell those products are interested in monitoring the current trend and analyzing people's interest. Such information can influence their marketing strategy or bring changes in the product design to meet the customers' needs.

Therefore, there is a growing need for algorithms and methods to automatically collect and process opinionated texts. Such methods are expected to classify the texts by their polarity (positive or negative), estimate the sentiments expressed and determine the opinion target and holder, where the target is the object or a subject of the opinion statement and the holder is usually the author of the text (but not limited to).

One of the basic tasks of sentiment analysis is classification of text polarity. Given a text, the system should determine whether its overall sentiment is negative or positive (or none, i.e. neutral). The general approach is to represent the text as a bag-of-words (or ngrams) with a binary weighting scheme, and use SVM for classification. Such a simple approach yields good results when provided with sufficient training data. Reference [1] reported 82.7 % accuracy on the movie review dataset. In their following work [2], the authors could improve the classification accuracy by adding a subjectivity detector to a preprocessing step, before the polarity classification, to remove objective sentences.

In this paper, we propose a simple technique for tuning the weighting scheme that improves the classification accuracy. Our technique is based on the normalization of term weights by their average term frequency across the document collection. The motivation behind this procedure is based on an observation that terms expressing an author's attitude to the described topic are unique in a document. While other terms that are not important for the classification decision are more frequent. Thus, if we divide a term's weight by its average term frequency, we prune low important words (such as articles, personal and possessive pronouns, etc.) and give more weight to unique keywords in a text. It turns out that review authors express their opinion trying to use a rich vocabulary and therefore words related to a sentiment would occur rarely in a text.

Another issue which can also be addressed by our normalization procedure is the lowering of the weights associated to the Named Entities (NE) which are strongly related to an opinion target (e.g. the name of actors, cast and producers in case of movie review dataset).

While these terms are very important for the overall classification accuracy, they are a source of bias for minor reviews. For example, if a dataset contains 10 positive reviews about the movie "Avatar", then it is very likely that the keywords "Avatar", "James Cameron" (the movie director), "Sam Worthington",

[2] http://twitter.com
[3] http://facebook.com

and "Zoe Saldana" (the movie cast) would cause any other review containing these keywords to be considered as a positive too. As a result, if a dataset contains an 11^{th} negative review about "Avatar", it would be probably misclassified as positive.

Thus we can lower the importance of NEs by normalizing theirs weights with the average term frequency across the corresponding Opinion Entity Document Set (OEDS)[4].

In the next section, we give a brief overview of prior works in sentiment analysis and research on improving polarity classification accuracy. In Sect. 3, we describe our normalization technique. Section 4 presents the data we have used for the validation. We provide details about our experimental setup and results in Sect. 5. Section 6 holds a description of our normalization technique for NE features to improve the polarity classification of minor reviews. In Sect. 7, we report on experiments with a second normalization technique for minor reviews. Finally, we conclude on our work in Sect. 8.

2 Related Work

An early work by [1] on polarity classification using bag-of-words model and machine learning reported 82.7 % accuracy. The authors found that using unigram features with binary weights yielded the highest accuracy.

In a follow up work, [2] augmented the classification framework with an additional preprocessing step, during which the sentences are first being classified as subjective or objective. The authors translated subjectivity classification into a graph-partitioning problem and used the min-cut max-flow theorem to solve it. Finally, the sentences labeled as "subjective" are extracted and passed to a general polarity classifier (bag-of-words model with SVM). The reported statistically significant improvement of the classification accuracy was from 82.8 % to 86.4 %.

Reference [3] used appraisal theory to produce additional features to be used in the classification. The authors built a taxonomy of appraisal and used it to identify appraisal groups within a text, such as "extremely boring" or "not that very good". For each appraisal group, a frame with 5 slots is filled up. The slots identifiers are: Attitude, Orientation, Force, Focus, and Polarity. Combinations of the first 3 of these slots were used to generate a feature vector. When backed up with a bag-of-words based classifier, the proposed method yielded 90.2 % accuracy, and 78.3 % standalone.

Reference [4] focused on a problem of the bag-of-word model, the information loss when representing a text by a non-related terms, thus losing the information contained in word order and syntactic relations between words in a sentence. To solve this problem, the authors proposed new features: word subsequences and dependency subtrees. Word subsequences were defined as a sequence of words

[4] The set of documents expressing opinions about the same opinion target (the same subject), for example, all reviews about the AVATAR movie represent an Opinion Entity Document Set.

obtained from a sentence by removing zero or more words. Dependency subtrees were obtained by extracting a part of a dependency tree, a sentence representation where nodes represent words and edges represent syntactic relations between words. Efficient mining algorithms were then used to find frequent subsequences and subtrees in the dataset. The combination of the proposed features with traditional n-gram features yielded 92.9 % classification accuracy on the movie dataset.

Reference [5] took a different approach to increase the accuracy of sentiment classification. Instead of adding supplementary features or text preprocessing steps, they focused on the words weighting scheme. The authors presented delta tf-idf weight function which computes the difference of a word's tf-idf score in a positive and a negative training sets. They claimed that the proposed technique boosts the importance of words unevenly distributed between the positive and the negative classes, thus these words should contribute more in the classification. Evaluation experiments on three different datasets showed statistically significant improvement of the classification accuracy. They achieved 88.1 % accuracy on the movie dataset.

Reference [6] performed a thorough study on different weighting schemes and the impact on the sentiment analysis systems' performance. In their study, the authors have tested different variations of the classic tf-idf scheme on three datasets: movie reviews, product reviews, and blog dataset. The best results were yielded by a variation of smoothed delta tf-idf. In the experimental setup of leave-one-out cross validation, the polarity classification accuracy on the movie and the product review datasets was 95–96 % depending on the scoring function variant used.

3 Proposed Method

3.1 Weighting Schemes

We will now give the formula describing our normalization function and present the rationale behind it.

Given a text T as a set of terms:

$$\mathrm{T} = \{t_1, t_2, \ldots, t_k\} \tag{1}$$

we define a feature vector of T as

$$\mathrm{tf}^{\mathrm{w}} = \{w(t_1), w(t_2), \ldots, w(t_k)\} \tag{2}$$

where $w(t_i)$ is a weight function of a term t_i (Table 1). We define a normalized feature vector as

$$\mathrm{tf}^{\mathrm{w}}_{\mathrm{n}} = \{\frac{w(t_1)}{n(t_1)}, \frac{w(t_2)}{n(t_2)}, \ldots, \frac{w(t_k)}{n(t_k)}\} \tag{3}$$

where $n(t_i)$ is a normalization factor of a term t_i (Table 2).

Table 1. A list of term weight functions

Notation	Equation
binary (bin)	1 (if $t_i \in T$, 0 otherwise)
term frequency (tf)	$\mathrm{tf}(t_i)$
inverse document frequency (idf)	$\log \frac{D}{\mathrm{df}(t_i)+1}$
term frequency inverse document frequency (tf-idf)	$\mathrm{tf}(t_i) \cdot \log \frac{D}{\mathrm{df}(t_i)}$
delta inverse document frequency (Δidf)	$\log \frac{D_n \cdot \mathrm{df_p}(t_i)+1}{D_p \cdot \mathrm{df_n}(t_i)+1}$
delta term frequency inverse document frequency (Δtf-idf)	$\mathrm{tf}(t_i) \cdot \log \frac{D_n \cdot \mathrm{df_p}(t_i)+1}{D_p \cdot \mathrm{df_n}(t_i)+1}$

Table 2. A list of normalization factors

Notation	Equation
none (1)	1
average term frequency (avgtf)	$\mathrm{avg.tf}(t_i)$
square of average term frequency (avgtf2)	$\mathrm{avg.tf}^2(t_i)$

In this research, we test the following list of weight functions:

Our proposed normalization function is based on a term's average frequency:

$$\mathrm{avg.tf}(t_i) = \frac{\sum_{\forall T, t_i \in T} \mathrm{tf(t_i)}}{|\forall T, t_i \in T|} \tag{4}$$

Thus, we compare the following list of normalization factors:

For example, a traditional binary weighting scheme [1] with our notation would look as follows:

$$\mathrm{tf}_1^{\mathrm{bin}} = \{\mathrm{bin}(t_1), \mathrm{bin}(t_2), \ldots, \mathrm{bin}(t_k)\} \tag{5}$$

With our proposed normalization factor:

$$\mathrm{tf}_{\mathrm{avgtf}}^{\mathrm{bin}} = \{\frac{\mathrm{bin}(t_1)}{\mathrm{avgtf}(t_1)}, \ldots, \frac{\mathrm{bin}(t_k)}{\mathrm{avgtf}(t_k)}\} \tag{6}$$

Delta tf-idf [5] with normalization factor of square avg.tf:

$$\mathrm{tf}_{\mathrm{avgtf2}}^{\Delta\mathrm{tf-idf}} = \{\frac{\Delta\mathrm{tf} - \mathrm{idf}(t_1)}{\mathrm{avgtf}^2(t_1)}, \ldots, \frac{\Delta\mathrm{tf} - \mathrm{idf}(t_k)}{\mathrm{avgtf}^2(t_k)}\} \tag{7}$$

Our normalization function is based on an observation that review authors tend to use rich vocabulary when expressing their attitude towards a movie or a product. Thus, the terms related to the sentiment expression (such as "outstanding") have average term frequency close or equal to 1. While other non-subjective

Table 3. Top-20 unigrams ordered by Δtf-idf (on the left) and normalized Δtf-idf (on the right) from the movie dataset. Unigrams related to movie or person names are highlighted bold.

Word	Avg.tf	Word	Avg.tf
ideals	1.33	conveys	1.08
frances	1.36	detract	1.09
comforts	1.00	criticized	1.00
supports	1.00	notoriety	1.00
ideology	1.20	ideal	1.33
gattaca	4.20	outstanding	1.06
outstanding	1.06	weaknesses	1.27
criticized	1.00	ideology	1.20
elmore	1.33	brisk	1.00
hawthorne	1.22	avoids	1.00
downside	1.00	judges	1.00
lebowski	6.88	slip	1.00
cunning	1.25	**frances**	1.36
gripping	1.06	**hawthorne**	1.22
judges	1.00	astounding	1.00
gretchen	1.38	scholars	1.00
unravel	1.00	discussion	1.00
burbank	2.00	hers	1.00
linney	1.38	abstract	1.00
niccol	2.38	obstacle	1.13

terms have higher average term frequency. Those include movie names, actors, brands and product parts as they are mentioned several times within texts.

The proposed delta tf-idf is supposed to target the same problem, filtering out general terms and favor the terms that are distributed unevenly in polarity sets. However, it creates a bias when a movie or a product name appears more often in positive (or negative) reviews.

We illustrate this example in Table 3, where a top-20 unigrams are shown with the number of positive and negative reviews they appear in and an average term frequency. In the left part of the table, we show unigrams ordered by delta tf-idf and in the right part ordered by delta tf-idf normalized by average frequency. As we can see, in the left part there are more unigrams related to a movie name or a person name (actors or movie-makers). For example, a movie name "gattaca" has a high rank according to delta tf-idf, because it has 19 positive reviews and no negative ones. The average term frequency of the term "gattaca" is 4.2, thus we can lower the importance of this term by normalizing its weight with the average term frequency (avg.tf).

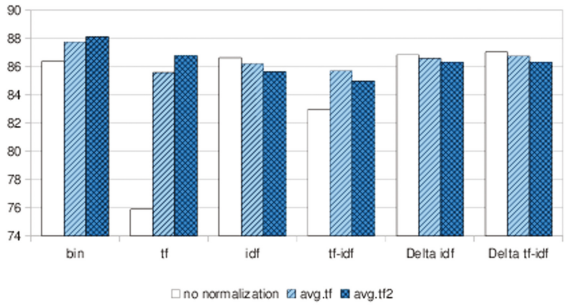

Fig. 1. Reported accuracy (in %) on the movie review dataset for document polarity classification across different weighting schemes and normalization factors.

4 Experimental Setup

To test our approach, we used two different datasets. The first is the movie review dataset[5], that has been first used in [1] and then in other research on sentiment analysis. The reviews are divided into a positive and a negative sets, each containing 1000 documents.

The second dataset consists of product reviews[6] from Amazon and was first used in [7]. The dataset is divided into 4 categories of products: books, DVDs, electronics, kitchen&housewares. Each category contains 1000 positive and 1000 negative reviews. This dataset is considered multi-domain as the contained documents are about various topics, as compared to the movie dataset which covers only the domain of movies.

We used an open source implementation of SVM classifier from the LIBLINEAR package [8] with default parameters and linear kernel. Each of the two datasets was evaluated separately using 10-fold cross validation. For our main evaluation criteria, we measured the average polarity classification accuracy (Fig. 1).

5 Results

The results of the reported accuracy on the movie reviews and the product reviews datasets, for document polarity classification, are presented in Tables 4 and 5.

First, we observe the same results as [1] reported on the movie review dataset: binary features outperform term frequency and tf-idf. We also observe the advantage of the delta tf-idf weighting scheme. Without any normalization, delta tf-idf yields the highest accuracy on the movie review dataset: 87.5 %.

[5] http://www.cs.cornell.edu/people/pabo/movie-review-data/
[6] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

Table 4. Reported accuracy on the movie review dataset for document polarity classification. The maximum accuracy is highlighted with a bold font.

Weight func.	Normalization		
	none	avg.tf	avg.tf2
bin	86.40	**87.75**	**88.15**
tf	75.90	85.60	86.80
idf	86.65	86.20	85.65
tf-idf	83.00	85.70	85.00
Δidf	86.85	86.60	86.35
Δtf-idf	**87.05**	86.75	86.35

Table 5. Reported accuracy on the product review dataset for document polarity classification. The maximum accuracy is highlighted with a bold font.

Weight func.	Normalization		
	none	avg.tf	avg.tf2
bin	78.77	79.36	79.71
tf	77.37	79.50	80.05
idf	79.06	79.02	79.10
tf-idf	78.66	79.00	78.76
Δidf	80.43	**80.76**	**80.83**
Δtf-idf	**80.69**	80.62	80.69

We further observe that our normalization technique improves the performance of binary (bin), term frequency (tf) and tf-idf weighting schemes. The accuracy of the tf scheme improves drastically, from 75.9 % to 86.80 % with avg.tf2 normalization. The bin scheme with the normalization yields the best results on the movie review dataset: 88.15 %. The accuracy of idf, Δidf and Δtf-idf schemes with the normalization decreases slightly.

The general observation from the product review evaluation is that cross-domain sentiment analysis is a more difficult task, as we can see in the results. The average performance of all the approaches are around 79–80 % as compare to 85–87 % on movie reviews. Similar results are obtained as in the movie review evaluation: Δtf-idf outperforms binary scheme, which in its turn outperforms term frequency and yields slightly better results than tf-idf. The normalization boosts the performance of the tf scheme, from 77.37 % to 80.05 %, such that it even outperforms the binary (79.71 %) and tf-idf schemes (79 %) with and without the normalization. The best result 80.83 % is yielded by the normalized Δidf which is slightly better than its version not normalized.

6 Named Entity Weighting for Improving Minor Opinion Polarity Classification

Most of the opinion mining models consider three elements, the opinion expression, the source and the target [9]. We are here interested by the opinion target. It is often referred to by means of Named Entities (person, product organization name, or location) in conjunction with the occurrence of a set of associated Named Entities, like for instance the cast or the film director, when the target is a movie.

In the classical approach used for opinion mining and polarity analysis, i.e. supervised machine learning with n-gram features [10], the system is often biased towards the majority opinion expressed in the training data. In particular, the NE n-grams used to refer to the opinion target are identified by the system as clues for the majority opinion, exactly in the same way as the specific vocabulary for expressing this opinion.

For instance, if we consider a film that has been a success like *AVATAR*, it is not only the mention of its title that would trigger a positive review classification, but also the presence of the name of the film director *James Cameron*, and this even for rarely occurring negative reviews.

Furthermore, NEs are not part of the general vocabulary used to express opinions and sentiments. In our mind an opinion mining system should be able to distinguish between the clues given by the explicit expression of opinion, and the clues associated to contextual features like NEs. The latter could be used in a second stage decision process for choosing the final opinion class. The advantage would be for the system first to be able to properly classify minority opinions and second to provide a justification about its classification decision in terms of either the language used in the opinion expression or the presence of contextual features, like particular NEs. To this end, we propose a second normalization weighting schemes to lower the weights of NEs. Hereafter, if a term (t_i) is recognized as a Named Entity, we compute its weight using a normalization function based on an intra opinion entity term's average frequency[7]:

$$\text{intra.oe.avg.tf}(t_i)^{NE} = \frac{\sum_{\forall D_{oe}, t_i \in D_{oe}} \text{tf}(\text{t}_\text{i})}{|\forall D_{oe}, t_i \in D_{oe}|} \tag{8}$$

Where D_{oe} is an element of the Opinion Entity Document Set for an Opinion Entity OE. As each D_{oe} is associated to a single OE entity and as review authors use NEs to describe an OE, the average term frequency of a NE across its corresponding D_{oe} is higher than the one computed over the whole corpus (Table 6).

[7] Computed over the Opinion Entity Document Set, i.e. the set of documents expressing opinions about the same opinion target, for example, all reviews about the AVATAR movie.

Table 6. Example of movie reviews extracted from the Imdb dataset. Terms NEs are highlighted bold.

Entity	Reviews
AVATAR	More of the **James Cameron** genius.
	Kudos to **Cameron**, **Avatar** is one of the (if not The) movie of the year.
	James Cameron's **Avatar** is the most entertaining and enthralling cinematic experiences of my life.
Star Wars	**George Lucas** enjoys an almost god-like status among sci-fi/fantasy fans worldwide.
	Not to mention **John Williams'** wonderful score, without of it, the movie wouldn't have been this great it's a perfect mix, that's what it is!

Table 7. Characteristics of preprocessed movie review dataset.

Initial number of reviews		Training and test sets sizes	
pos	neg	train	test
25000	25000	3680	1580

7 Experiments

7.1 Data

For our purpose, we have split into training and test set the Large Movie Review Dataset [11] used for our experiments in a special way. The 50,000 texts have been spread evenly between an equal proportions of negative and positive opinions in both the training set and the test set, following the procedure first proposed in [10]. For each movie, the number of reviews has been fixed both for training and testing. We took 3 documents of each movie for test and 7 for training. Characteristics of both datasets are presented in Table 7. These numbers were chosen heuristically in order to maximize the total number of reviews.

To separate a dataset into training and test sets, first, we group all the reviews by their entity (movie) identified by a unique ID in the dataset. Next, we select groups that have enough numbers of positive and negative reviews. From the selected groups, from each entity we select all the reviews of a dominant polarity in this group and move them to the training set. The remaining reviews from each group are moved to the test set. We call this dataset "minor biased", because the test set contains reviews with minor polarities. We expect traditional settings for polarity classifiers to yield worse results on this dataset due to the bias in reviews for each product. To prove that the drop of performance is caused effectively by the biased features, we construct a dataset composed of the same reviews but reorganized, such that the reviews in the test set for each entity have the same polarity as the dominant polarity in the training set for each entity. We call

Table 8. Classification accuracy obtained using different normalization schemes on movie reviews.

	unb.	Δ	minb.	Δ	majb.	Δ
Bigrams + binary						
no	79.6		71.9		83.5	
avg.tf	79.7	+0.1	72.8	+0.9	84.0	+0.5
avg.tf.intra.oe.avg.tf	**81.5**	**+1.9**	**76.3**	**+4.4**	84.2	+0.7
Bigrams + Delta tf-idf						
no	83.0		69.9		87.6	
avg.tf	82.9	−0.1	76.0	+6.0	86.1	−1.5
avg.tf.intra.oe.avg.tf	**84.2**	**+1.2**	**78.3**	**+8.5**	85.8	−1.8

this dataset "major biased", because the test set contains reviews with major polarities. Finally, we compose the "unbiased" dataset, by separating reviews such that entities in the test set have no reviews in the training set. Named Entities were tagged with TAGMATICA, which is an industrial strength Named Entity tagger [12].

7.2 Results

First, we prove the negative effect of entity specific features on classification accuracy of minor reviews. We ran experiments on 3 variants of the datasets: unbiased (unb), minor biased (minb), major biased (majb). We have used bigrams (bi) with binary (bin) and Delta tf-idf weights. Results on classification accuracy across the datasets and features are presented in Table 8. Notice that we cannot directly compare accuracy values across different variants of datasets, as they are composed of different test data. However, we assume that our datasets are homogeneous and results obtained with different dataset variants reflect the complexity of the classification task.

Impact of OE specific terms and NEs. Looking at Table 8, we see that Opinion-Entity-specific and Named Entities features cause performance drop on the minor biased set as compared to the unbiased set (unb vs. minb). We also observe a boost in performance on the major biased dataset in spite of a smaller training size (unb vs. majb). This shows that our classifier learns to associate OE-specific terms and NE features with the opinion entity major polarity, instead of learning the affective language model of opinion expression. Results are similar across different datasets, variants of datasets, and features. Delta tf-idf while improving overall accuracy, causes misclassification of minor review because it gives more importance to opinion entity-specific and Named Entities features. We can observe this by comparing the results of using Delta tf-idf on the minor biased set with the unbiased and major biased datasets.

Next, we have evaluated the effect of the proposed normalization schemes on classification accuracy. As we observe from the previous experiments, normalizing

NE weights with the $intra.oe.avg.tf()$ increases the performance, see the highlighted locations in Table 8.

8 Conclusion

We have proposed two techniques to tune the weighting scheme of a general polarity classifier. The first technique is generic and based on a normalization of a term's weight by its average term frequency. The proposed normalization method increases the importance of terms that are rare in a document. Thereby decreasing weights of frequent terms and therefore reducing a bias when an object has more positive (or negative) reviews. The second technique lowers the importance of Named Entities about opinion targets, by normalizing their weights in the feature vector representations used by classical n-gram. For the first technique, the experimental evaluations was performed on two datasets of different size, topic and homogeneity: movie and product review dataset. Both evaluations showed that the proposed normalization method increases the performance of binary, term frequency, and tf-idf weighting schemes. The performance of the term frequency is increased significantly (from 75.9 % to 86.80 % on the movie review dataset and from 77.37 % to 80.05 % on the product reviews). The normalized binary scheme yielded the highest observed classification accuracy (88.15 %) on the movie review dataset. The normalization of Δtf-idf scheme improves slightly its performance on the product review dataset, however our scheme can be used when Δtf-idf is not available. For example, when there is no data split in two sets, or when there are more sets than two (i.e. more sentiment classes: positive, negative, neutral). In this case, a binary weighting scheme with the proposed normalization method should be used.

For the second technique, although the performance improvement is not as important for bigram models (+1.5 %) with unbiased training datasets, it is nevertheless positive which proves that our NE weighting scheme performs as well as classical methods. But the evaluation experiments performed on especially organized versions of standard datasets showed large improvement in classification accuracy of minor reviews (+8.5 %), which proves that our NE weighting scheme impacts positively the classification accuracy of minor reviews, essential for weak signals detection and early opinion trend reversal detection.

References

1. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, EMNLP '02, pp. 79–86. Association for Computational Linguistics, Morristown (2002)
2. Pang, B., Lee, L.: A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, ACL '04. Association for Computational Linguistics, Stroudsburg (2004)

3. Whitelaw, C., Garg, N., Argamon, S.: Using appraisal groups for sentiment analysis. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, CIKM '05, pp. 625–631. ACM, New York (2005)
4. Matsumoto, S., Takamura, H., Okumura, M.: Sentiment classification using word sub-sequences and dependency sub-trees. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 301–311 (2005)
5. Martineau, J., Finin, T.: Delta TFIDF: an improved feature space for sentiment analysis. In: Proceedings of the Third AAAI Internatonal Conference on Weblogs and Social Media. AAAI Press, San Jose (2009)
6. Paltoglou, G., Thelwall, M.: A study of information retrieval weighting schemes for sentiment analysis. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, ACL '10, pp. 1386–1395. Association for Computational Linguistics, Morristown (2010)
7. Blitzer, J., Dredze, M., Pereira, F.: Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification. In: Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 440–447. Association for Computational Linguistics, Prague (2007)
8. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: a library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)
9. Paroubek, P., Pak, A., Mostefa, D.: Annotations for opinion mining evaluation in the industrial context of the doxa project. In: Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC). ELDA, Valetta (2010)
10. Pak, A.: Automatic, Adaptive, and Applicative Sentiment Analysis. Ph.D. thesis, Thèse de l'École Doctorale d'Informatique de l'Université Paris-Sud, Orsay, June 2012
11. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the ACL, pp. 142–150. ACL, Portland (2011)
12. Francopoulo, G., Demay, F.: A deep ontology for named entities. In: Proceedings of the International Conference on Computational Semantics, Interoperable Semantic Annotation Workshop, ACL (2011)