# Improving classification of tweets using linguistic information from a large external corpus

Hugo Lewi Hammer, Anis Yazidi, Aleksander Bai, and Paal Engelstad

Department of Computer Science
Oslo and Akershus University College of Applied Sciences
Norway

hugo.hammer@hioa.no, anis.yazidi@hioa.no,
aleksander.bai@hioa.no, paal.engelstad@hioa.no

**Abstract.** The bag of words representation of documents is often unsatisfactory as it ignores relationships between important terms that do not co-occur literally. Improvements might be achieved by expanding the vocabulary with other relevant word, like synonyms.

In this paper we use word-word co-occurence information from a large corpus to expand the vocabulary of another corpus consisting of tweets. Several different methods on how to include the co-occurence information are constructed and tested out on the classification of real twitter data. Our results show that we are able to reduce the number of erroneous classifications by 14% using co-occurence information.

keywords: *classification, co-occurrence information, text mining, tweets*

## 1 Introduction

Founded in 2006, Twitter (www.twitter.com) has grown to become one of the most popular social media services, known for its 140-character restriction on each post. In addition to a large general user base, Twitter is used extensively by celebrities, politicians, and news services to entertain, engage, or inform their followers. With over 500 million users, Twitter sees a daily stream of more than 400 million tweets a day [1].

Twitter is known to be an important source for early detecting of important events like breaking news, changes in the stock market, spread of diseases, earthquakes etc or analyzing different trends in politics, fashion, entertainment etc, see e.g. [2–6]. Such approaches are typically based on training a machine learner on a bag-of-words representation of the tweets, maybe in addition to other features like number of words, publication time etc. The bag of words representation is often unsatisfactory as it ignores relationships between important terms that do not co-occur literally. Many important words and phrases for correct classification may never occur in the training material, but only show up in the test material (e.g. future tweets). A bag-of-words approach will not be able to detect such tweets since the important words never occurred in the training set. For example, suppose we want to detect tweets about the war in Syria. In the manually annotated training material we may have good predictors like “al-Assad”, “Syria”, “Homs” etc, but may miss other relevant phrases like “Damascus”, “gas attack”, “Baath party”, “ISIL” which potentially could improve the classifications since such words are likely to occur in future tweets about the Syrian war.

In this paper we suggest to “enrich” the vocabulary in the training material with other potentially relevant phrases by using word-word co-occurrence information from an other large news corpus (1.1 billion words). Computing words that tend to co-occur with “al-assad” in the news corpus, we find among the top ten words “bashar”, “al-sharaa” (vice president in Syria), “negotiations” and “syria” which seem like other relevant words to detect tweets about the Syrian war. It’s not obvious what’s the best way to incorporate such external co-occurrence information in the training material of tweets. We suggest a large set of different approaches and test them extensively on real twitter data.

The paper is extention of the preliminary results in the paper [7].

## 2 Related work

Techniques for enriching text fall under two main categories: those who use intrinsic information contained in the current corpus and those who use extern resources. A representative example of intrinsic techniques is the the Self-Term Expansion Methodology due to Pinto et al. [8] for clustering tweets. The method compromises two main steps: the Self-Term Enriching step, and a Term Selection step. The Self-Term Enriching procedure enriches the text representation of the tweets by exploiting the current tweets corpus and without the need of any external corpus, that is why the technique is called Self-Term Enriching. Terms of a documents are represented with a set of co-related terms. A co-occurrence list is calculated from the target data set by applying Pointwise Mutual Information (PMI). The Term Selection step identifies the most important features and tries to reduce the noise introduced by the the Self-Term Enriching phase.

The second category of techniques for enriching text representation uses external resources other than the current text materials to be clustered or classified. It is worth mentioning that the later techniques have received most attention in the literature compared to techniques that resort to intrinstic information for the enriching task. For example in [9–12], the authors enrich the text representation using WordNet [13] where terms of the documents are replaces with their hypernym and synonym.

Similarly, the seminal work of Gabrilovich et al. [14] leverages knowledge bases from Wikipedia and Open Directory Project (ODP) in order to enhance the textual representation of short messages. The authors concluded that augmented knowledge based features generated from ODP and Wikipedia improved the text categorization task.

Alahmadi et al. [15] use an approach based on supplementing the bag-of-words representational scheme with a concept-based representation that utilises Wikipedia as a knowledge base.

In [16] Wikipedia semantic knowledge are used to tackle data sparseness in a question answering task. Experiments show that the approach significantly outperforms the baseline method (with error reductions of 23.21%).

Chen et al. [17] propose a word-word co-occurrence matrix based method for improved relevance feedback in information retrieval. Unlike other studies about word association, the authors consider the influence of the inter word distance and co-windows ratio. Experiments with TREC dataset demonstrate the effectiveness of the method.

## 3 Word-word co-occurrence matrix and Document term matrix

In this section we represent relevant background information for the rest of the paper. More specifically we define the word-word co-occurrence matrix (COM) and the document term matrices (DTM).

### 3.1 Word-word co-occurrence matrix

Suppose we have a large corpus consisting of a total of $N$ words and let $w_1, w_2, \ldots, w_{N_w}$ denote the different unique words in the corpus. Further let $N_i,\ i \in \{1, 2, \ldots, N_w\}$ denote the number of times $w_i$ occurs in the corpus and let $N_{ij},\ i, j \in \{1, 2, \ldots, N_w\}$ denote the number of times $w_i$ occurs in the neighbourhood of $w_j$ in the corpus. The neighbourhood of a word, $w_j$, is typically those words closest to $w_j$ in front and behind in the text. We assume symmetry such that $N_{ji} = N_{ij}$. A COM is the matrix with the element $N_{ij}$ in position $(i, j),\ i, j = 1, 2, \ldots, N_w$. A COM computed from a large corpus is a highly valuable tool to analyze semantic relations between words, see e.g. [18, 19].

Suppose we want to use COM to compute the semantic relation between $w_i$ and $w_j$. There are typically three main approaches.

**Correlation** The empirical correlation in occurrence with other words

$$\text{Corr}(w_i, w_j) = \frac{\text{Cov}(w_i, w_j)}{\sqrt{\text{Var}(w_i)\text{Var}(w_j)}}$$

where

$$\text{Var}(w_i) = \frac{1}{N_w - 1}\sum_{k=1}^{N_w}(N_{ik} - \overline{N_{i\cdot}})^2$$

$$\text{Cov}(w_i, w_j) = \sum_{k=1}^{N_w}(N_{ik} - \overline{N_{i\cdot}})(N_{jk} - \overline{N_{j\cdot}})$$

where

$$\overline{N_{i\cdot}} = \frac{1}{N_w} \sum_{k=1}^{N_w} N_{ik}$$

**Angle** The angle between the co-occurrence vectors for $w_i$ and $w_j$ given as $[N_{i1}, N_{i2}, \ldots, N_{i\,N_w}]$ and $[N_{j1}, N_{j2}, \ldots, N_{j\,N_w}]$.

**PMI** The Pointwise Mutual Information between $w_i$ and $w_j$ defined as

$$PMI(w_i, w_j) = \log_2 \frac{P(w_i, w_j)}{P(w_i)P(w_j)} = \log_2 \frac{P(w_i \mid w_j)}{P(w_i)} = \log_2 \frac{P(w_j \mid w_i)}{P(w_j)} \quad (1)$$

Looking at the rightmost expression the numerator denotes the probability that $w_j$ occurs in the neighborhood of $w_i$ and the denominator the probability that a randomly selected word from the corpus is $w_j$. If $w_i$ and $w_j$ tend to co-occur, we expect that $P(w_j \mid w_i) > P(w_j)$.
The PMI can be estimated as follows based on the rightmost expression in (1)

$$\widehat{PMI}(w_i, w_j) = \frac{N_{ji}/N_i}{N_j/N}$$

Because of the symmetry, $N_{ji} = N_{ij}$, we get the same expression for $\widehat{PMI}(w_i, w_j)$ estimating based on the leftmost and middle expression in (1). Also note that $\widehat{PMI}$ are just simple reweightings of the entries in COM and thus very fast to compute.

In our experiments the PMI performed better than the two other approaches and the descriptions below therefore are based on PMI.

### 3.2 Document term matrix

Other words for a document term matrix (DTM) are bag-of-words and n-grams. Suppose that a corpus consist of $D$ tweets (more generally documents). Let $n_{di}$ denote the number of times word $w_i$ occur in tweet $d \in \{1, 2, \ldots, D\}$ and $n_w$ the total number of unique words in the $D$ tweets. A DTM is the matrix with the elements $n_{di}$ in positions $(d, i),\ d = 1, 2, \ldots, D, i = 1, 2, \ldots, n_w$. A natural generalization is to not only use words, but all phrases of subsequent words in the corpus called n-grams. In this paper we only resort to single words (unigram). Reweightings of the pure term frequencies in a DTM is also very common, e.g. the TF-IDF ([18], chapter 15).

## 4 Incorporating co-occurrence information from a large external corpus in a document term matrix

In this section we present different methods to incorporate COM information from a large external corpus to a DTM. We start by expanding the vocabulary of DTM from all the unique words in the tweets to the union of the unique words in the tweets and the words in COM. See Fig. 1 for a simple visualization of the expansion. The gray part shows the additional words added to the original DTM shown as the white part of the matrix. Our goal is to add reasonable values in the gray part of the matrix and adjust values in the white part of the matrix to improve classification. To simplify the notation below, let $r_{ij}$ refer to $\widehat{PMI}(w_i, w_j)$. Also assume that all words in the tweet vocabulary are part of the COM vocabulary. In practice we obtained this by letting words that is in the tweet vocabulary and not in the COM vocabulary, are added to COM with all co-occurence frequencies with other words equal to zero.

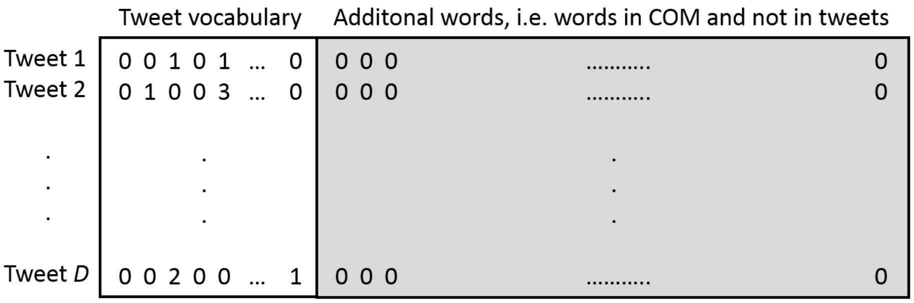

**Fig. 1.** Illustration of the expansion (shown in gray) of the original tweet DTM shown in white.

Suppose a tweet $d \in \{1, 2, \ldots, D\}$ consists of the $\eta_d$ unique words $w_{d(1)}, \ldots, w_{d(\eta_d)}$ and recall that we assume that all being part of the COM vocabulary $w_1, w_2, \ldots, w_{N_w}$. Further let $n_{d,d(1)}, \ldots, n_{d,d(\eta_d)}$ denote the frequency (or some reweighting like TF-IDF) of $w_{d(1)}, \ldots, w_{d(\eta_d)}$. Define the matrix $\text{PMI}_{\text{tweet}}$ consisting of the entries $r_{d(i),j}$, $i = 1, 2, \ldots, \eta_d$, $j = 1, 2, \ldots, N_w$ containing the PMI scores between the words in the tweet and all the words in COM. Fig. 2 illustrates this matrix. Based on $\text{PMI}_{\text{tweet}}$ we can expand the vocabulary of the tweet $d$ in different ways. Maybe the most natural is for each word in COM to compute the sum of PMI scores for the words in the tweet and add this values to the expanded DTM shown in Fig. 1

$$\widetilde{n}_{d,j} = \frac{1}{\eta_d} \sum_{i=1}^{\eta_d} r_{d(i),j}, \quad j = 1, 2, \ldots, N_w \tag{2}$$

where $\widetilde{n}_{d,j}$ refers to the scores to add to the expanded DTM in position $(d, j)$. We can interpret this as adding the (pointwise mutual) information from all the tweet words together and thus the approach intuitively seem reasonable .

$$\begin{array}{c|cccc|}
 & w_1 & w_2 & \cdots\cdots & w_{N_w} \\
\hline
w_{d(1)} & r_{d(1),1} & r_{d(1),2} & \cdots\cdots & r_{d(1),N_w} \\
w_{d(2)} & r_{d(2),1} & r_{d(2),2} & \cdots\cdots & r_{d(2),N_w} \\
\vdots & & & \vdots & \\
w_{d(\eta_d)} & r_{d(\eta_d),1} & r_{d(\eta_d),2} & \cdots\cdots & r_{d(\eta_d),N_w} \\
\hline
\end{array}$$

**Fig. 2.** Illustration of the of the matrix $\mathrm{PMI}_{\mathrm{tweet}}$.

A natural modification of (2) is to weight with the occurrences of the tweet words

$$\widetilde{n}_{d,j} = \frac{1}{\eta_d} \sum_{i=1}^{\eta_d} n_{d,d(i)} \, r_{d(i),j} \tag{3}$$

For words being part of the original tweet vocabulary, we update the values from $n_{d,d(i)}$ to $n_{d,d(i)} + \alpha\widetilde{n}_{d,d(i)}$ for $i \in 1, 2, \ldots, \eta_d$ where $\alpha$ is a rescaling parameter since we have no reason to believe that $n_{d,d(i)}$ and $\widetilde{n}_{d,d(i)}$ are on the same scale. For the other words we substitute zero with
$\alpha\widetilde{n}_{d,j}$, $j \in \{1, 2, \ldots, N_w\}\backslash\{d(1), d(2), \ldots, d(\eta_d)\}$ where $A\backslash B$ refers to all the elements in $A$ except those that are in $B$. Below we describe different alternatives to (3).

A challenging part of including PMI-information is that large amounts of noise may be included. Consider the following fictive tweet about the Syrian war: "President al-Assad agrees to negotiate". Using the method in (3), we add the PMI scores for all the words together, but words like "agrees" and "to" are not at all relevant for the classification to Syrian war and may introduce unfortunate noise. Intuitively we expect that words that have high PMI score for two or more words in the tweet are more likely to be relevant words, while words with only one high PMI value are less likely to be relevant. For example, the word "al-Sharaa" (vice president in Syria) has high PMI score with *all* the three words "President", "al-Assad" and "negotiate" and thus is most likely be a relevant word. We can achieve this property by doing transformation with monotonically increasing concave functions like the logarithm and $r^\gamma, \gamma < 1$. The following simple example motivates such transformations. Suppose a word $w_i$ has a PMI score of 8 with two of the words in the tweet (and zero to the other words) and suppose that another word $w_j$ has a PMI score of 16 to one word in the tweet (and zero to the other words). Using (3), the two words will get equal score, but transforming with the logarithm $w_i$ gets the score $\log_2 8 + \log_2 8 = 3 + 3 = 6$ and $w_j$ gets the $\log_2 16 = 4$. We see that $w_i$ now gets a higher score then $w_j$ after the transformation. Therefore we generalize (3) to

$$\widetilde{n}_{d,j} = \frac{1}{\eta_d} \sum_{i=1}^{\eta_d} n_{d,d(i)} (r_{d(i),j})^{\gamma} \tag{4}$$

and setting $\gamma < 1$ we achieve what is explained above. If $r_{d(i),j} < 0$ in (4), we replace $(r_{d(i),j})^{\gamma}$ with $-(|r_{d(i),j}|)^{\gamma}$. In the rest of the paper we denote the approach based on (4) for method SPMI. The methods below are variants of this.

- Method RAW: We take sum of the raw ratios in (1) instead of the PMIs

$$\widetilde{n}_{d,j} = \frac{1}{\eta_d} \sum_{i=1}^{\eta_d} n_{d,d(i)} (2^{r_{d(i),j}})^{\gamma} = \frac{1}{\eta_d} \sum_{i=1}^{\eta_d} n_{d,d(i)} 2^{\gamma\, r_{d(i),j}} \tag{5}$$

- Method RAWL: We take the logarithm of the scores $\widetilde{n}_{d,j}$ from method RAW.
- Method MAXL: In stead of taking the sum over PMI values as for the methods SPMI, RAW and RAWL, we just use the maximal PMI score. Let $\hat{i} = \arg \max_{i \in 1,2,\ldots \eta_d} \{r_{d(i),j}\}$ and compute the scores as

$$\widetilde{n}_{d,j} = n_{d,d(\hat{\imath})} r_{d(\hat{\imath}),j}$$

- Method MAX: The same as method MAX except that we use the raw ratios

$$\widetilde{n}_{d,j} = n_{d,d(\hat{\imath})} 2^{r_{d(\hat{\imath}),j}}$$

- Method GEOM: The geometric mean. If we compute $\widetilde{n}_{d,j}$ from (3), the geometric mean can be computed as

$$2^{\widetilde{n}_{d,j}}$$

While method SPMI with a low value for $\gamma$ ends up with many scores $\widetilde{n}_{d,j}$ with almost the same values, method RAW combined with a high value of $\gamma$ ends up with only a few scores $\widetilde{n}_{d,j}$ with high values and all the other close to zero (after proper rescaling with the parameter $\alpha$). The other methods above are different variants which lies between these extremes.

### 4.1 Only important words

As mentioned above, a potential challenge with the idea of using COM information is that the scores $\widetilde{n}_{d,j}$ from the methods above may be disturbed by PMI information from irrelevant words like "agree" and "to" from the example tweet above. Another approach to reduce the possible noise is to only include PMI information for words with a positive correlation with the response. For the example tweet we expect that words like "al-Assad", "president" and "negotiate" are positively correlated with tweets about the Syrian war compared to tweets about other topics while words like "agree" and "to" are less correlated with the Syrian war. A more detailed description of how to achieve this is as follows. Assume we have a multiclass classification problem with classes $C_1, C_2, \ldots, C_K$ and let $Y_{dk}, d = 1, 2, \ldots, D, k = 1, 2, \ldots, K$ be equal to one if tweet $d$ belong to class $k$ and zero else. Further let $Y_k = [Y_{1k}, Y_{2k}, \ldots, Y_{Dk}]$. We now compute the correlation between the $Y_k$ and word frequencies of a word $w_i$ being part of the tweet vocabulary

$$\mathrm{Corr}(Y_k, w_i) = \frac{\mathrm{Cov}(Y_k, w_i)}{\sqrt{\mathrm{Var}(Y_k)\mathrm{Var}(w_i)}}$$

where

$$\mathrm{Cov}(Y_k, w_i) = \sum_{d=1}^{D} (Y_{dk} - \overline{Y_k})(n_{di} - \overline{n_{\cdot i}})$$

We now only include PMI information for tweet words where the maximal correlation to $Y_1, Y_2, \ldots, Y_K$ is above some threshold $\tau$. This result in the following rewriting of equation (4)

$$\widetilde{n}_{d,j} = \frac{1}{\eta_d} \sum_{i=1}^{\eta_d} I\left( \max_{k=1,2,\ldots,K} \{\mathrm{Corr}(Y_k, w_{d(i)})\} > \tau \right) n_{d,d(i)} (r_{d(i),j})^{\gamma} \qquad (6)$$

and similar for the other methods above. $I(\cdot)$ is the indicator function.

## 5 Linguistic resources

In this section we present the resources necessary to evaluate the methods above. All the analyzes are done for the Norwegian language.

The COM are computed from a huge corpus that is made openly available by the National Library of Norway (NLN). The corpus consists of news articles collected from Norwegian newspapers from 1998 until 2011. This corresponds to roughly 1.1 billion Norwegian words distributed over 4 million articles. To compute $N_{ij}$, we used a neighborhood of six words in front and behind of $w_j$ (recall Section 3.1). We only used words that occurred at least 50 times in the news corpus ending up with a vocabulary with 287904 unique words.

The Twitter corpus is selected from all tweets published in Norwegian on Twitter from 20th of July to 8th of August 2011 a total of about two million tweets. We selected a subset of tweets as follows:

1. We counted the number of times different hashtags were used.
2. Among the most frequently used hashtags we manually picked hashtags related to six topics as summarized in Table 1 and selected all the tweet consisting at least one of these hashtags.

The resulting corpus consists of a total of 21270 tweets. Since the tweets are from the time span around the tragic 22th July terror it is as expected that we observe many tweet related to this. The classification task is to classify the correct topic of these tweets when all the hashtags are removed from the tweets.

**Table 1.** Details of the six topics. Columns from left to right: Description of cluster, number of tweets for the different topics and hashtags representing the topics.

| Topic | No. | Hashtags |
|---|---|---|
| The 22th July 2011 terror | 11519 | #Utøya #PrayForOslo etc |
| Justin Beiber | 3409 | #Bieber #Bieberlove etc |
| Norwegian national elections | 2218 | #Valg #Valg11 etc (valg = election) |
| Tour de France | 1668 | #TdF #2dF etc |
| The Øya music festival | 1311 | #Øya |
| Libya | 879 | #Libya |

## 6 Experiments

In this section we compare the classification performance of the different extended DTMs described above. We base our classifications on multinomial LASSO regression [20]. Multinomial LASSO regression can document excellent classification performances, and is at least as effective as Support vector Machine [21]. Another advantage of the LASSO is that the estimated parameter space is sparse, i.e. only a little handful of the hundreds of thousands of words in the vocabulary are used by the classifier. This gives us the possibility to inspect which words are used by the classifier and interpret the results. We return to this in Section 6.2.

In all the experiments we sat the rescaling parameter $\alpha = 1$ (recall Section 4). Our results show that expanding the tweet vocabulary using only important words (Section 4.1) in stead of all the words, improved classification performance. Thus in Section 6.1, we only present results based on the important words method in Section 4.1. In the experiments we sat $\tau = 0.08$ in (6) ending up with a total of 316 important words fairly evenly spread over the six classes in Table 1.

We expect that incorporating external information is particularly useful if the number of documents (tweets) in the annotated training material are few. Then many important predictors (words) are missing in the training material and thus not being part of the classifier. Our results is in accordance with this. Using 30% or more of the tweet corpus to train the classifier (more than 6381 tweets), the reduction in erroneous classifications is below 5% compared to not using external information. Using less than 30% of the tweets to train the classifier, the reduction in erroneous classifications is between 5 and 15%.

### 6.1 Classification performance

Above we summarized the main results from the experiments. In this section we look closer at the cases which gave the best result, i.e. we expanded only by important words (Section 4.1) and used less than 30% of the tweet corpus in the training set. Table 2 shows classification results using 5% (1064 tweets) and 10% (2127 tweets) of the tweets for training. To reduce the uncertainty in our results in Table 2, we used cross validation repeatedly using different parts of

**Table 2.** The values represent the percentage of tweets classified to the correct class. Columns two to four show results using 5% of the tweets as a training set, while the last three columns show results for 10% training. NOEXT refers to classification using the tweet DTM (no additional words included).

| | 5% training | | | 10% training | | |
|---|---|---|---|---|---|---|
| Method | $\gamma = 0.1$ | $\gamma = 1$ | $\gamma = 10$ | $\gamma = 0.1$ | $\gamma = 1$ | $\gamma = 10$ |
| NOEXT | – | 69.0 | – | – | 73.1 | – |
| SPMI | 73.1 | 68.9 | 73.4 | 75.8 | 72.4 | 75.5 |
| RAW | 72.9 | 68.3 | 72.9 | 75.8 | 72.4 | 75.5 |
| RAWL | 72.6 | 73.3 | 73.3 | 75.6 | 75.8 | 75.9 |
| MAX | – | 72.9 | – | – | 75.5 | – |
| MAXL | – | 73.3 | – | – | 75.9 | – |
| GEOM | – | 72.2 | – | – | 75.1 | – |

**Table 3.** The table shows the words that are the best predictors (translated from Norwegian to English) for the different topics. The words in the second column (tweet words) were part of the original vocabulary of the tweets in the training set. Third column shows words that are good predictors but were not part of the original tweet vocabulary but added using some of the methods in Section 4.

| Topic | Tweet words | Added words |
|---|---|---|
| The 22th July 2011 terror | auf, dead, killed, people, norway, people, police, together, thoughts, sad, utøya | arrested, armed, bomb |
| Justin Beiber | album, bieber, love, follow, hope, justin, culture, 4ever, aloooot, follower, lookin | dream, teenagers, brands, selena, girls, loves |
| Norwegian national elections | frp, nrk, political, jensen, vote, tv2, election, voting machine | argument, conservative, industry, vice chairman, prime minister |
| Tour de France | schleck, boasson, edwald, stage, france, paris, tdf, jersey | meters, astana, eneco, stages, french, prestigious, overall lead, fell |
| The Øya music festival | fleet foxes, håkan, kanye, consert, music, sunshine, plays | winehouse, gitarist, linkin, mayhem, acquired, surferosa, equalized |
| Libya | gaddafi, jail, hell, hotel, fire, libya, libyan, nightmare, tripoli | attacks, continued, kaim, coastal, town, sirte, officer, nato, regime, soldiers |

the tweets as training and test corpus. The results in Table 2 are the average classification performance for the different cross validation runs. The width of confidence intervals based on the cross validation runs where about 0.1, i.e. in practice it is no uncertainty in the values in Table 2.

As expected a higher percentage of the tweets are classified correctly when 10% of the tweets are used for training compared to only 5%. For 10% training the highest reduction in erroneous classifications were

$$\frac{(100 - 73.1) - (100 - 75.9)}{100 - 73.1} \cdot 100\% = 10.4\%$$

For 5% training the highest reduction is

$$\frac{(100 - 69.0) - (100 - 73.4)}{100 - 69.0} \cdot 100\% = 14.2\%$$

We see, as expected, that when the training set is small inclusion of external co-occurence information have a larger positive effect on the classification performance. An other interesting observation is that NOEXT using 10% training performs poorer (73.1%) than using 5% training and external information (73.4%). In other words it is better to include external co-occurence information than increasing the number of annotated tweets from 5% (1064 tweets) to 10% (2127 tweets). Having in mind that manual annotation of documents are very resource demanding, this is quite an impressive result and documents the usefulness of the methods in this paper.

From Table 2 we see that the most natural choice of using external information, SPMI with $\gamma = 1$ (which is equivalent to (3)) do not document any improvements. It seems that better results are achieved by either using a few words with high values (e.g. RAW and $\gamma = 10$ and MAX) or many words with almost the same value (e.g. SPMI and $\gamma = 0.1$), not the alternatives in between.

### 6.2 Words used in the classifier

In this section we inspect which words are the best predictors of the different classes in Table 1. Using the multinomial LASSO model with a higher value of the regularization parameter than the optimal value, classification is performed using only a few words. Naturally the classification performance is reduces using such a sparse classifier, but on the other hand interpretation is very easy. Table 3 shows the results for the case with 5% of the tweets in the training set and method SPMI with $\gamma = 10$ and a high value of the regularization parameter. We see that both the tweet words (second column) and the added words (third column) are very relevant words for the different topics. E.g for the Justin Bieber topic, words like his ex girlfriend Selena are not part of the original training vocabulary, but added using the methods in Section 4. Selena occurred several times in the test set and thus improved classification were achieved by including Selena. For the Øya music festival, we see that several other relevant artists are added as extra words like (Amy) Winehouse, Mayhem and Surferosa. For the Libya topic words like the Deputy Foreign Minister (Khaled) Kaim and references to the battle of Sirte also were good predictors resulting in improved classification performance.

## 7 Closing remarks

In this paper we show how external information from a word-word co-occurrence matrix can be used to improve the classification of tweets. The methods in this paper are particularly useful if the number of tweets in the training set is small. E.g. if the number of tweets is about a thousand, our results show a reduction in erroneous classifications with about 15%.

There are several interesting directions for further research using word-word co-occurrence information. We believe that the constructed methods in this paper are useful also for other sorts of documents and could be interesting to investigate further. It could also be interesting to evaluate the methods above for unsupervised tasks like clustering and topic modeling [22].

## References

1. Zubiaga, A., Spina, D., Martinez, R., Fresno, V.: Real-time classification of twitter trends. Journal of the American Society for Information Science and Technology **66**(3) (2015) 462 – 473
2. Petrović, S., Osborne, M., Lavrenko, V.: Using paraphrases for improving first story detection in news and twitter. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. NAACL HLT '12, Stroudsburg, PA, USA, Association for Computational Linguistics (2012) 338–346
3. Zhang, X., Fuehres, H., Gloor, P.A.: Predicting Stock Market Indicators Through Twitter I hope it is not as bad as I fear. Procedia - Social and Behavioral Sciences **26** (2011) 55–62
4. Lampos, V., De Bie, T., Cristianini, N.: Flu detector: Tracking epidemics on twitter. In: Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part III. ECML PKDD'10, Berlin, Heidelberg, Springer-Verlag (2010) 599–602
5. Lee, K., Palsetia, D., Narayanan, R., Patwary, M.M.A., Agrawal, A., Choudhary, A.: Twitter trending topic classification. In: Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops. ICDMW '11, Washington, DC, USA, IEEE Computer Society (2011) 251–258
6. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: Real-time event detection by social sensors. In: Proceedings of the 19th International Conference on World Wide Web. WWW '10, New York, NY, USA, ACM (2010) 851–860
7. Hammer, H.L., Yazidi, A., Bai, A., Engelstad, P., et al.: Improving classification of tweets using word-word co-occurrence information from a large external corpus. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing, ACM (2016) 1174–1177
8. Pinto, D., Rosso, P., Jiménez-Salazar, H.: A self-enriching methodology for clustering narrow domain short texts. The Computer Journal **54**(7) (2011) 1148–1165
9. Hotho, A., Staab, S., Stumme, G.: Ontologies improve text document clustering. In: Data Mining, 2003. ICDM 2003. Third IEEE International Conference on, IEEE (2003) 541–544
10. Hotho, A., Staab, S., Stumme, G.: Wordnet improves text document clustering. In: Proc. SIGIR Semantic Web Workshop. (2003)

11. Rodriguez, M., Hidalgo, J., Agudo, B.: Using wordnet to complement training information in text categorization. In: Proceedings of 2nd International Conference on Recent Advances in Natural Language Processing II: Selected Papers from RANLP. Volume 97. (2000) 353–364
12. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: Proceedings of the 12th international conference on World Wide Web, ACM (2003) 519–528
13. Miller, G.A.: Wordnet: A lexical database for english. Commun. ACM **38**(11) (1995) 39–41
14. Gabrilovich, E., Markovitch, S.: Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In: AAAI. Volume 6. (2006) 1301–1306
15. Alahmadi, A., Joorabchi, A., Mahdi, A.: A new text representation scheme combining bag-of-words and bag-of-concepts approaches for automatic text classification. In: GCC Conference and Exhibition (GCC), 2013 7th IEEE, IEEE Press (2013)
16. Cai, L., Zhou, G., Liu, K., Zhao, J.: Large-scale question classification in cqa by leveraging wikipedia semantic knowledge. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management. CIKM '11, New York, NY, USA, ACM (2011) 1321–1330
17. Chen, Z., Lu, Y.: A word co-occurrence matrix based method for relevance feedback. Journal of Computational Information Systems **7**(1) (2011) 17 – 24
18. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge, MA, USA (1999)
19. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors forword representation. `http://nlp.stanford.edu/projects/glove/glove.pdf` (2015) [Online; accessed 27-July-2015].
20. Friedman, J., Hastie, T., Tibshirani, R.: Regularization Paths for Generalized Linear Models via Coordinate Descent. Journal of Statistical Software **33**(1) (2010) 1–22
21. Genkin, A., Lewis, D.D., Madigan, D.: Large-Scale Bayesian Logistic Regression for Text Categorization. Technometrics **49**(14) (2007) 291–304
22. Blei, D.M.: Probabilistic topic models. Commun. ACM **55**(4) (2012) 77–84