Supervised Tweet Classification

Josip Milić, Tomislav Marinković, Domagoj Pereglin

University of Zagreb, Faculty of Electrical Engineering Computing
Unska 3, 10000 Zagreb, Croatia
{josip.milic,tomislav.marinkovic, domagoj.pereglin}@fer.hr

Abstract

In this paper we describe our implementation of a method for topic classification of Twitter messages written in Croatian language into one of six predefined classes. We present the results of three machine learning algorithms (SVM, logistic regression, and k-NN) used for classification. Our training data consists of Croatian twitter messages acquired through Twitter API. These messages are manually labeled and passed to classifiers which are trained and then evaluated.

1. Introduction

Twitter[1] is a popular microblogging service where users post messages called *tweets*. These messages can contain different media types, but usually are consisted of different forms of text. Twitter's most valued differentiating feature from other social media services is the mandatory brevity because the size of each message is limited to 140 characters. This makes it a great platform for sharing information in a short and straightforward form such as user comments and news (or at least an essential part of it). Each Twitter user can subscribe and read tweets from other users which are usually their friends or people who they admire and value their opinions. Users can also be official representatives of news agencies and post news tweets. Many of them are specialized for different kinds of news (e.g. information technology, politics), but by subscribing to them users can also expect possibly unwanted information such as advertisements or tabloid news. Tweet classification could be used for filtering tweets related to topics specified by the user. For the purpose of demonstration, topics were news (segmented as IT, politics, sports and general news) and deals (e.g. job offers, user ads).

Several Croatian Twitter users were chosen because of their language and expected types of tweets (e.g. @bugonline for IT news, @hrtsport for sports news) and their tweets were acquired via Twitter API.

In order to train a classifier, supervised learning usually requires hand-labeled training data. Six descriptive labels were used each for every topic, detailed in subchapter 3.2. Labeling.

In grammar, inflection is the modification of a word to express different grammatical categories such as tense, case, voice, gender, and other. Croatian language is highly inflective so special care was taken regarding this matter in preprocessing step. For feature extraction we used the bag-of-words model. This is a simplifying representation where text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and word order. It is commonly used in methods of document classification where the frequency of occurrence of each word is used as a feature for training a classifier.

[1] http://www.twitter.com

2. Related works

There are several research papers regarding tweet classification.

One approach to short text classification is to use a small set of domain-specific features. Siriam et al. extracted these features from the author's profile and text (Sriram et al., 2010). Their set of features consisted of one nominal and seven binary features. Nominal feature is author of a tweet, and binary features are shortened words and slangs, time-event phrases, emphasized words, currency and percentage signs, opinioned words, "@username" at the beginning of the tweet, and "@username" within the tweet. They classified tweets to a predefined set of generic classes such as News, Events, Opinions, Deals, and Private Messages. Naïve Bayes classifier is trained using 5-fold cross validation.

Batool et al. analyzed extracted information from tweets using keyword based knowledge extraction (Batool, 2013). They extracted keywords, entities, synonyms, and parts of speech from tweets which are then used for classification and sentiment analysis. The extracted knowledge was enhanced using domain-specific seed based enrichment technique in a way that decreases information loss. They reported improvement from 0.1% to 55% with their proposed system.

3. Data and methods

Implemented method consists of five stages: gathering data, labeling, preprocessing, feature extraction, and machine learning.

3.1. Gathering data

Croatian tweets were acquired by using GrepTweet [2]. It retrieved tweets from chosen users as text files where each file row had form *tweet ID* | *posting date* | *text content*. Tweet IDs were used for differentiation and labeled tweet texts were used for model training and testing.

3.2. Labeling

Table 1 shows chosen descriptive category (class) names and a number of labeled tweets per each category.

[2] http://greptweet.com/

Text Analysis and Retrieval 2016: Course Project Reports (TAR 2016), pages 43–46, University of Zagreb, Faculy of Electrical Engineering and Computing, Zagreb, June 2016.

category	number of labels
NEWS_TECHNOLOGY	1043
NEWS_POLITICS	1657
NEWS_SPORT	1052
NEWS_REST	2738
DEALS	976
REST	3257
Σ	10723

Table 1: Category names and number of labels per category

Tweets containing news were split into four subcategories and tweets containing deals refer to all kinds of ads and offers, ranging from job offers to item selling ads. Tweets that do not belong to any of these categories were labeled as REST.

Tweets were manually labeled by the authors. Each tweet received a single label. Tweets with multiple topics were labeled as their primary topics or were discarded as REST if the author couldn't decide. Cohen's kappa was used to measure inter-rater agreement. It measures the agreement between two annotators who each classify N items into C mutually exclusive categories. The equation for kappa is: $\kappa = \frac{p_0 - p_e}{1 - p_e}$, where p_o stands for relative observed agreement and p_e for hypothetical probability of chance agreement.

Each annotator provided 50 labeled tweets from each category. Calculated values are shown in Table 2.

	$Annotators(\kappa)$		
	$A1$	$A2$	$A3$
$A1$	1.000	0.742	0.745
$A2$	0.742	1.000	0.805
$A3$	0.745	0.805	1.000

Table 2: Cohen's kappas between annotators

Kappas for all pair combinations of annotators were summed and averaged to get mean Cohen's kappa. Calculated mean Cohen's kappa is $\bar{\kappa} = 0.764$.

3.3. Preprocessing

Before classification, retrieved tweet texts were preprocessed. Firstly, tweet texts were purified by removing punctuation marks (e.g. commas, brackets, quotation marks, stroke) which were considered bad for classifier training. Secondly, TakeLab preprocessor was used for NLP pipeline preprocessing of tweet text. The NLP pipeline consists of sentence segmentation $\triangleright$ tokenization $\triangleright$ POS tagging $\triangleright$ morphological processing (stemming and lemmatization).

Sentence segmentation and tokenization splits sentences into smaller meaningful parts. Lemmatisation is a process of transforming a word to its basic form.

3.4. Feature extraction

TF-IDF is the weight computed as the product of the term frequency component and the inverse document frequency component.

$$tf(k_i, d_j) = 0.5 + \frac{0.5 * freq(k_i, d_j)}{max(freq(k, d_j)|k \in d_j)} \quad (1)$$

$$idf(k_i, D) = log \frac{|D|}{|d \in D|k_i \in d_j|} \quad (2)$$

TF-IDF value (product of TF and IDF values) reflects how important a word is to a document in a collection or corpus. Training and test datasets were converted to corresponding TF-IDF vectors by using created vocabulary of words retrieved from the training dataset. The created vocabulary consisted of 14822 words.

For the purpose of "helping" the classifiers, map of special words for each category was created. Each TF-IDF vector is enhanced by adding six (one for each category) values at the end of the vector. Each value represents the size of intersection of a set of special words from category and current set of words. Evaluation results were slightly improved with using only dozens of special words per category.

4. Machine learning and evaluation

Dataset was split into train and test parts (train dataset = 70%). Three classifiers were used for tweet classification. Two baseline classifiers were used for comparing evaluated results. Main classifiers were: SVM, Logistic Regression and k-NN. Statistical values used for classifier evaluation ($True, False, Positive, Negative$ denoted as T, F, P, N) are:

- $precision = \frac{T_P}{T_P + F_P}$, the fraction of retrieved instances that are relevant
- $recall = \frac{T_P}{T_P + F_N}$, the fraction of relevant instances that are retrieved
- $F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{2 \cdot T_P}{2 \cdot T_P + F_P + F_N}$, the harmonic mean of precision and recall
- $accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}$, the proportion of true results (both true positives and true negatives) among the total number of cases examined

Cross validation was performed on the whole dataset (training and test labeled data combined). The dataset was split randomly to train and test data twenty times for classifier fitting/testing purpose.

4.1. Dummy classifier

Dummy classifier is a classifier that makes predictions using simple rules. Two dummy classifiers were used for baseline classification. The second gave single prediction for all test data - most common category in dataset.

category	$precision$	$recall$	$F1 - score$
NEWS_TECH	0.000	0.000	0.000
NEWS_POLITICS	0.000	0.000	0.000
NEWS_SPORT	0.000	0.000	0.000
NEWS_REST	0.000	0.000	0.000
DEALS	0.000	0.000	0.000
REST	0.293	1.000	0.453
Σ	0.049	0.167	0.075
Accuracy	29.282 %		

Table 3: Evaluated dummy classifier values

4.2. SVM classifier

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning) the algorithm outputs an optimal hyperplane which categorizes new examples.

Implemented SVM is using linear kernel because of the number of features. SVM classifier was fitted with labeled and preprocessed training data represented as TF-IDF vectors. For the purpose of optimal classification, penalty parameter C was chosen from predetermined values with grid search method. Optimal C for training data is $C = 0.7$.

category	$precision$	$recall$	$F1 - score$
NEWS_TECH	0.736	0.650	0.691
NEWS_POLITICS	0.871	0.844	0.857
NEWS_SPORT	0.902	0.856	0.879
NEWS_REST	0.577	0.597	0.586
DEALS	0.851	0.795	0.822
REST	0.687	0.732	0.709
Σ	0.771	0.746	0.757
Accuracy	72.61 %		

Table 4: Evaluated SVM classifier values

Calculated cross validated accuracy with standard deviation is $accuracy_{CV} = 68.24(\pm 9)\%$.

4.3. Logistic regression classifier

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution.(Freedman, 2009)

Logistic regression classifier was fitted with labeled and preprocessed training data represented as TF-IDF vectors. For the purpose of optimal classification, inverse of regularization strength C was chosen from predetermined values with grid search method. Optimal C for training data is $C = 12.75$.

category	$precision$	$recall$	$F1 - score$
NEWS_TECH	0.773	0.626	0.692
NEWS_POLITICS	0.869	0.835	0.852
NEWS_SPORT	0.9126	0.834	0.871
NEWS_REST	0.564	0.604	0.583
DEALS	0.865	0.788	0.825
REST	0.680	0.738	0.708
Σ	0.777	0.737	0.755
Accuracy	72.27 %		

Table 5: Evaluated logistic regression classifier values

Calculated cross validated accuracy with standard deviation is $accuracy_{CV} = 68.02(\pm 9)\%$.

4.4. k-NN classifier

k-Nearest Neighbors algorithm (k-NN) is a non-parametric method used for classification. k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms.

k-NN classifier was fitted with labeled and preprocessed training data represented as TF-IDF vectors. For the purpose of optimal classification, number of neighbors k was chosen from predetermined values with grid search method. Optimal k for training data is $k = 28$.

category	$precision$	$recall$	$F1 - score$
NEWS_TECH	0.797	0.374	0.509
NEWS_POLITICS	0.859	0.681	0.759
NEWS_SPORT	0.910	0.744	0.819
NEWS_REST	0.682	0.376	0.485
DEALS	0.565	0.752	0.645
REST	0.515	0.843	0.639
Σ	0.721	0.628	0.643
Accuracy	63.35 %		

Table 6: Evaluated k-NN classifier values

Calculated cross validated accuracy with standard deviation is $accuracy_{CV} = 63.07(\pm 7)\%$.

5. Conclusion

Tweet classification evaluation values of used classifiers were satisfiable considering the language of text and used preprocessing methods and tools. Equivalent version of WordNet (lexical database) designed for Croatian language would be of a great help for classification of tweets. There is a number of things which can be done to improve results such as enlargement of labeled dataset (use of supervised annotating of large scale would certainly help), including more special features (with more intelligent approach than just counting the length of the intersection), and further tweaking of used classifiers.

Web application for demonstration of a practical use of tweet classification was created. Its basic idea is that users can filter tweets by choosing desired categories. Only those tweets which were classified into desired classes are shown on screen. For the purpose of experimenting, classifier selection and manual input of text is also included.5.

By improving the tweet classifier, users could be less exposed to tweets considered as noise and irrelevant, but also wouldn't miss tweets from desired categories misclassified as undesired categories.

References

Maqbool Batool, Khattak. 2013. Precise tweet classification and sentiment analysis. IEEE.

David A. Freedman. 2009. Statistical models: Theory and practice. page 128. Cambridge University Press.

B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. 2010. Short text classification in twitter to improve information filtering. In *33rd international ACM SIGIR conference*, pages 841–842. ACM.

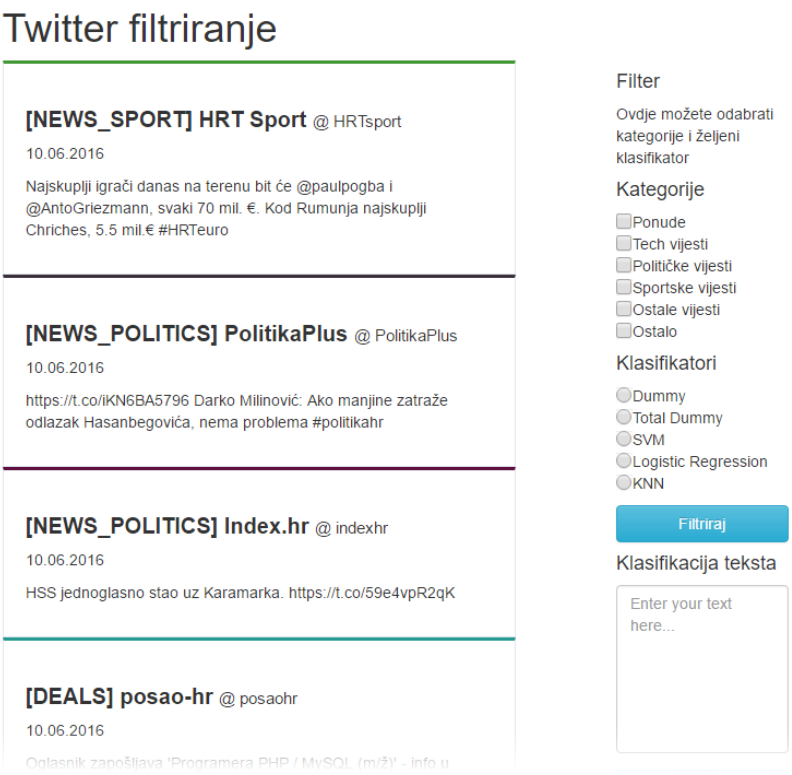

Figure 1: Web page of tweet filtering web application