10.12871/clicit2014219

Self-Evaluating Workflow for Language-Independent Sentiment Analysis

Arseni Anisimovich
Minsk State Linguistic University, Minsk, Belarus[1]
WorkFusion Inc., New York, USA[2]
arseni.anisimovich@gmail.com[1]
aanisimovich@workfusion.[2]

Abstract

English. This paper describes a generic framework that relies on extra-linguistic features of text as well as on its content to perform sentiment analysis in four different dimensions. Routine described in the paper allows not only extraction of opinion mining data but also describes a framework for continuous relearning of Support Vector Machines classifiers in order to improve classification results when dataset size is increased or new parameters of classifier are found to be of better quality.

Italiano. *Questo articolo descrive una tecnica generale che si basa su caratteristiche extra-linguistiche del testo, e anche sul suo contenuto, allo scopo di eseguire una sentiment analysis in quattro dimensioni. Questo procedimento non solo permette l'estrazione dei dati di sentiment analysis, ma descrive anche un algoritmo di ri-apprendimento continuo con support vector machines (particolarmente utile nei casi in cui ci sono ulteriori esempi o nuovi parametri che migliorano la qualità dell'analisi).*

1 Introduction

The rise of new media especially social ones have brought absolutely new source of up-to-date information on different topics that can be exploited in different tasks. One of such tasks is opinion mining or sentiment analysis that could bring vital information to many researchers including, but not limited to sociologists, campaigners, and marketing analysts.

Sentiment analysis of English texts has drawn scholars' attention about a decade ago (Turney, 2002; Pang et al., 2002) and provided basic experimental data and directions of research for scientific community. That resulted in annual shared tasks and conferences that bring attention to the problem and raise the bar for the state-of-the-art approaches on a regular basis.

However, the information to be analyzed in modern world does not include sole English texts. That fact has inspired raising interest in developing mechanisms for sentiment analysis of texts in languages other than English (Basile et al., 2014). While some scholars propose the focus on leveraging resources from languages with more data (Mihalcea et al., 2007), this paper describes a generic approach in sentiment analysis that can be applied to any collection of labelled data without preliminary linguistic work.

2 System Description

Sentiment analysis, as the task that this paper is aimed to solve, is a basic binary classification problem when treating each of sub-tasks (Positive and Negative Polarity, Subjectivity and Irony) as a separate problem.

Recent researches prove that in sentiment analysis as a classification task, Support Vector Machines (SVM) classifiers perform with a decent quality (Mullen, 2004), (Gamon, 2004). LibSVM (Chang, 2011) was used as an algorithmic implementation of SVMs.

Since libSVM comes with several Support Vector Machines types and several kernels, the workflow was set up to train all applicable classifiers with a ranging parameters to automatically find the best configurations for every classification task.

SVM's possibility to train a stable classifier on a limited set of labeled data has been of a huge help because of variable proportion of positive and negative examples of a class in each subtasks:

	Pos. Example	Neg. Example
Subjectivity	2804 (70.68%)	1163 (29.32%)
PolPositive	1132 (28.54%)	2835 (71.46%)
PolNegative	1729 (43.58%)	2238 (56.42%)
Irony	498 (12.55%)	3469 (87.46%)

Table 1. Amount of examples per subtask.

Despite the fact that positive and negative example ratio is different per task, training set was unified for every subtask as well as the features selection. The main ranging parameters were SVM parameters and feature frequency threshold.

Since results were only reported for constrained run, there was no external information used in the feature set. However, several simple text transformations were performed to facilitate classifier training basing on extra-linguistic knowledge.

2.1 Feature Selection

The assumption that the set of features is similar in all subtasks was made thus eliminating the need for several training set generation procedures. However, several transformations of raw tweet text were performed.

Firstly, all URLs were converted to a single word-marker '*url*' because of insignificancy of link address. Then, the presumption that some links bring more personalized information was token, and the URLs were classified into two groups: Long URLs and Shortened URLs. The former is a link in an unconstrained format peculiar to a specific website while the latter is provided by third-party service (e.g. Google URL Shortener[1], Bitly[2], or Twitter's internal service[3]).

The reason behind that transformation is that when an application (either way on a mobile device or in a browser) posts a link, it usually converts a given URL in short format (in order to save the space in a 140-symbol message), but, as the research of training dataset has shown, when a news agency posts a link, it usually posts it as-is, without any shortening service. Since the information whether the tweet belongs to an individual or to an organization is a valuable feature, this transformation was applied for every tweet and gave 2% average increase in terms of both precision and recall.

Another important transformation of dataset was to turn all the variety of smileys into information. From all the smileys only two categories were selected: those representing a sad emotion and those representing a happy emotion, since polarity task had only two dimensions and variety of emotions that can be represented using smileys is convertible to these two subsets.

Except of described transformations, size of tweet relative to maximum size of tweet in training dataset (in bytes) was added to raw text as well as quotation markers, uncertainty or fragmentary text markers (for example three dots), re-tweet markers, hashtag markers, and Twitter picture (pic.twitter.com) markers in order to catch all the information that not only exists outside of the language, but is a distinctive feature of modern Internet communication and its implementation (Twitter as a platform and its client applications as instruments). Described transformations may be applied to any tweet in any language and still will produce comparable amount of training information.

2.2 Vector Normalization

Since SVM is a vector-based classifier and requires a vector of values as input for both training and classification procedure, a binary vector for each document was built using token occurrence as a '1' value and token absence as '0'. Token is understood as a sequence of non-whitespace characters.

This approach is usual to SVM feature generation, however it lacks the information about number of occurrences of a token in the text, and if in the case of stop word this information will not give any classification weight at all, quantity of emotion markers or picture amount in the text are priceless information which might be the straw that may break the back of misclassification camel.

Since the value of every token was only 0 or 1, in the described approach token occurrence in a document was scaled with maximum token occurrence in the training dataset thus turning possible values of a single feature from binary 0/1

[1] https://goo.gl/

[2] https://bitly.com/

[3] http://t.co/

vector into vector of values 0..1 thus saving the information for classifier to train on.

SVM's vector nature was a huge gain when compared to probability-based classifiers, since if one class tends to have less token occurrences and in testing set there is even smaller amount of those, SVM will not turn that feature into non-relevant, but will do its best to correctly classify example by comparing incoming vector against trained hyper plane.

2.3 Feature Pruning

As it was mentioned earlier, amount of positive and negative examples for each dimension of sentiment analysis varies a lot, leading to great feature imbalance. One of the approaches that can be used to eliminate negative impact on sentiment analysis quality is feature frequency limitation mechanism that excludes from training and testing vector those features that occur less than a predefined threshold.

Despite the fact that there are approaches that exclude features on the basis of discriminative function pruning analysis (DFPA)(Mao, 2004) this paper sticks to examinations of options to select most corresponding minimal feature frequency suitable for each subtask. Optimal parameters vary greatly, for example:

	PolNegative Precision	**PolPositive Precision**
FeatFreq: 15	35,46%	57,85%
FeatFreq: 4	38,82%	49,32%

Table 2. Precision changes over feature frequency parameter selection.

Automatic routine of choosing best parameters allows not only find best values for current task with current dataset, but also, if a researcher has access to continually growing dataset, existing models may be retrained in background with dataset growth and achieve better quality over new data.

2.4 Experimental Workflow

As it was said above, initial dataset for solving each of four subtasks is the same and when it comes into the system, training procedure begins from same starting point. Baseline of precision and recall is set using one-rule classifier (presuming that all examples should be classified as the majority of examples in training set).

Baseline is used to exclude those combinations of SVM types and kernel types that bring results worse than baseline (however, in this particular task, it never occurred and all applicable SVM classifiers were training all at once).

To eliminate the threat of biased testing set ten-fold cross-validation is used on every set of parameters during evaluation of classifier. Average of precisions and recalls for each cross-validation run is then used to rank set of parameters as most or least applicable to a given classification task.

Set of classifier parameters varies from SVM type and kernel type, and the only common parameter is feature frequency threshold. Experiments have shown that for the SENTIPOLC-2014 task for described approach following feature frequencies limits bring best results:

Irony	3
Subjectivity	15
PolPositive	3
PolNegative	7

Table 3. Feature frequencies thresholds per subtask.

These results correlate with common sense knowledge since both irony and positive attitude can be expressed in many ways and negative attitude, despite being expressed more often than positive attitude, lacks that variety of words to use. Limitations of Twitter message size and Internet slang provides a set of shorthands to express subjectivity and stay in the margins of tweet.

Different SVMs also train with different parameters specific to an algorithm, for example for linear SVMs the parameter *C* (cost parameter) was ranged from default 1 up to 100, for nu-SVC *ν* (nu) parameter was ranged from 0.01 up to 0.45 . Best parameters are selected for all the SVM and kernel types.

In the last step framework chooses best combination of feature frequency, SVM type and kernel type and trains final model on whole dataset to have a 'production' model that will be used to rank against testing data. In the SENTIPOLC-2014 task following parameters were chosen for each subtask:

Subtask	**FeatFreq**	**Classifier** (type/kernel)
Irony	3	c-SVC, linear (c=11)
Subjectivity	15	c-SVC, linear (c=11)
PolPositive	3	c-SVC, linear (c=9)
PolNegative	7	ν-SVC, linear (ν=0.43)

Table 4. Parameters of SVM classifiers.

All subtasks except for negative polarity were ranked using F1-measure while negative polarity was ranked using classification precision since basically, any F1-measure best classifier was one-rule classifier totally missing positive examples of negative polarity.

3 Conclusion

Described system didn't take first places in any constrained run task in SENTIPOLC-2014 shared task. However, resulting scores correlated with those obtained in cross-validation of 'production' classifiers while being 5-10% lower than development ones:

Subtask	**Expected**	**Real**	**Top**
Subjectivity 7/9	0.6545	0.5825	0.7140
Polarity 6/11	0.6812	0.6026	0.6771
Irony 3/7	0.5828	0.5394	0.5901

Table 5. Expected results with rankings.

Nonetheless, the approach presented in this paper has proven itself valid to be used against Twitter messages without any preliminary linguistic work. Features were independent from language of a tweet and all text transformations may be applied to a message in any language.

Described approach, unfortunately, lacks the information about syntactic structure of text of the tweet which may be eliminated or at least leveled with the help of a standard syntactic parser that should provide a uniform representation of syntactic structure for any language given, for example, dependency grammar tree.

In unconstrained run, there is a point of constant update of a training set using crowd sourcing platforms, which can provide data with high quality using initial training set not only as a classifier training set, but also as an example to teach crowd workers and maintain their quality as described in (Lease, 2011). That will give not only more complete dataset, but also will provide sources for relearning the classifier on new data that may reflect changes in the Internet slang that may occur in a split second.

References

Valerio Basile, Andrea Bolioli, Malvina Nissim, Viviana Patti, and Paolo Rosso. 2014. Overview of the Evalita 2014 SENTIment POLarity Classification Task. *Proceedings of the 4th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'14).*

Chih-Chung Chang, and Chih-Jen Lin. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*.

Michael Gamon. 2004. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. *Proceedings of the 20th international conference on Computational Linguistics*. ACL.

Matthew Lease. 2011. On Quality Control and Machine Learning in Crowdsourcing. *Human Computation*.

Mao, K.Z. 2004. Feature subset selection for support vector machines through discriminative function pruning analysis. *Systems, Man, and Cybernetics, Part B: Cybernetics*. Vol. 34, Issue 1. IEEE.

Rada Mihalcea, Carmen Banea, and Janyce Wiebe. 2007. Learning Multilingual Subjective Language via Cross-Lingual Projections. *Proceedings of the Association for Computational Linguistics (ACL).* pp. 976–983.

Tony Mullen, and Nigel Collier. 2004. Sentiment Analysis using Support Vector Machines with Diverse Information Sources. *EMNLP*. Vol. 4.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pp. 79–86.

Peter Turney. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *Proceedings of the Association for Computational Linguistics.* pp. 417–424.