JOINT_FORCES:
Unite Competing Sentiment Classifiers with Random Forest

Oliver Dürr, Fatih Uzdilli, and Mark Cieliebak
Zurich University of Applied Sciences
Winterthur, Switzerland
{dueo, uzdi, ciel}@zhaw.ch

Abstract

In this paper, we describe how we created a meta-classifier to detect the message-level sentiment of tweets. We participated in SemEval-2014 Task 9B by combining the results of several existing classifiers using a random forest. The results of 5 other teams from the competition as well as from 7 general-purpose commercial classifiers were used to train the algorithm. This way, we were able to get a boost of up to 3.24 F_1 score points.

1 Introduction

The interest in sentiment analysis grows as publicly available text content grows. As one of the most used social media platforms, Twitter provides its users a unique way of expressing themselves. Thus, sentiment analysis of tweets has become a hot research topic among academia and industry.

In this paper, we describe our approach of combining multiple sentiment classifiers into a meta-classifier. The introduced system participated in SemEval-2014 Task 9: "Sentiment Analysis in Twitter, Subtask–B Message Polarity Classification" (Rosenthal et al., 2014). The goal was to classify a tweet on the message level using the three classes positive, negative, and neutral. The performance is measured using the macro-averaged F_1 score of the positive and negative classes which is simply named "F_1 score" throughout the paper. An almost identical task was already run in 2013 (Nakov et al., 2013).

The tweets for training and development were only provided as tweet ids. A fraction (10-15%) of the tweets was no longer available on twitter, which makes the results of the competition not fully comparable. For testing, in addition to last year's data (tweets and SMS) new tweets and data from a surprise domain (LiveJournal) were provided. An overview of the provided data is shown in Table 1.

Using additional manually labelled data for training the algorithm was not allowed for a "constrained" submission. Submissions using additional data for training were marked as "unconstrained".

Dataset	Total	Pos	Neg	Neu
Training (Tweets)	8224	3058	1210	3956
Dev (Tweets)	1417	494	286	637
Test: Twitter2013	3813	1572	601	1640
Test: SMS2013	2093	492	394	1207
Test: Twitter2014	1853	982	202	669
Test: Twitter'14Sarcasm	86	33	40	13
Test: LiveJournal2014	1142	427	304	411

Table 1: Number of Documents we were able to download for Training, Development and Testing.

Our System. The results of 5 other teams from the competition as well as from 7 general-purpose commercial classifiers were used to train our algorithm. Scientific subsystems were s_gez (Gezici et al., 2013), s_jag (Jaggi et al., 2014), s_mar (Marchand et al., 2013), s_fil (Filho and Pardo, 2013), s_gun (Günther and Furrer, 2013). They are all "constrained" and machine learning-based, some with hybrid rule-based approaches. Commercial subsystems were provided by

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 366–369,
Dublin, Ireland, August 23-24, 2014.

Lymbix (c_lym), MLAnalyzer[1] (c_mla), Semantria (c_sem), Sentigem (c_snt), Syttle (c_sky), Text-Processing.com (c_txp), and Webknox (c_web). Subsystems c_txp and c_web are machine learning-based, c_sky is rule-based, and m_mla is a mix (other tools unknown). All subsystems were designed to handle tweets and further text types.

Our submission included a subset of all classifiers including unconstrained ones, leading to an unconstrained submission. The 2014 winning team obtained an F_1 score of 70.96 on the Twitter2014 test set. Our approach was ranked on the 12th place out of the 50 participating submissions, with an F_1 score of 66.79. Our further rankings were 12th on the LiveJournal data, 12th on the SMS data, 12th on Twitter-2013, and 26th on Twitter Sarcasm.

Improvement. Although our meta-classifier did not reach a top position in the competition, we were able to beat even the best single subsystem it was based on for almost all test sets (except sarcasm). In previous research we showed that same behaviour on different systems and data sets (Cieliebak et al., 2014). This shows that also other systems from the competition, even best ones, probably can be improved using our approach.

2 Approach

Meta-Classifier. A meta-classifier is an approach to predict a classification given the individual results of other classifiers by combining them. A robust classifier, which can naturally handle categorical input such as sentiments by design, is the random forest classifier (Breiman, 2001). The algorithm uses the outputs of individual classifiers as features and the labels on the training data as input for training. Afterwards, in the test phase, the random forest makes predictions using the outputs of the same individual classifiers. We use the random forest implementation of the R-package "randomForest" and treat the three votes (negative, neutral, positive) as categorical input.

Training Data. To build a meta-classifier, first, one has to train all the subsystems with a dataset. Second, the meta-classifier has to be trained based on the output of the subsystems with a different dataset than the one used for training the subsystems. We decided to take the natural split of the data provided by the organizers (see Table 1). For the scientific subsystems we used the Training set to train on; for training the random forest classifier we used the Dev set. The commercial systems were used "as-is", in particular, we did not train them on any of the provided data sets. Table 2 shows the performance of the individual subsystems on the different data sets.

ID	Dev	SMS2013	Twitter2013	Twitter2014	Sarcasm2014	LiveJournal2014
s_gez	32.22	31.23	30.77	28.57	**51.57**	50.83
s_jag	61.47	56.17	60.21	62.73	44.26	63.91
s_mar	28.95	22.94	26.68	22.86	31.01	24.47
s_fil	52.88	49.94	55.61	55.08	38.22	56.41
s_gun	**63.93**	**61.51**	**65.33**	**65.09**	48.80	**68.91**
c_lym	48.38	44.40	48.68	54.17	34.87	58.71
c_mla	49.79	46.41	50.17	47.74	43.16	59.02
c_sma	55.89	52.26	56.15	53.51	49.33	56.53
c_sky	56.30	52.04	54.67	56.28	40.60	54.61
c_txp	43.69	46.47	41.15	44.00	**59.74**	56.57
c_web	47.44	41.64	45.21	48.83	45.25	53.45
c_snt	**56.86**	**58.42**	**62.17**	**58.35**	36.08	**65.74**

Table 2: F_1 scores of the individual systems. Bold shows the best commercial or scientific system per data set; grey cells indicates the overall maximum.

3 Experiments

There exist three obvious selections of subsystems for our meta-classifier: all subsystems, only scientific subsystems, and only commercial subsystems (called All_Subsystems, All_Scientific, and All_Commercial, respectively). Table 3 shows performance of these selections of subsystems on the data sets. For comparison, the table shows also the performance of the overall best individual subsystem in the first row. It turns out that All_Subsystems is almost always better than the best individual subsystem, while the other two meta-classifiers are inferior.

Testing All Subsets. We performed a systematic evaluation on how the performance depends on the choice of a particular selection of individual subsystems. This resembles feature selection, which is a common task in machine learning, and

[1] mashape.com/mlanalyzer/ml-analyzer

	Dev (OOB)	SMS2013	Twitter2013	Twitter2014	Twitter2014 Sarcasm	LiveJournal2014
Best Individual	*63.93*	*61.51*	*65.33*	*65.09*	*48.80*	*68.91*
All_Subsystems	63.54	**64.22**	67.03	67.70	**46.37**	71.11
All_Scientific	64.52	60.42	64.54	64.99	43.35	67.86
All_Commercial	62.11	58.34	60.70	63.86	44.85	65.57
Max_OOB_Subset	**68.27**	63.02	**67.49**	**68.33**	45.40	**71.43**
Our Submission	65.00	62.20	66.61	66.79	45.40	70.02

Table 3: Performance (in F1 score) of meta-classifiers with different subsystems. The subset used in our submission is composed of s_gez, s_jag, s_mar, s_fil, s_gun, c_sma, c_sky, c_snt. "Max_OOB_Subset" is composed of s_jag, s_mar, s_gun, c_lym, c_sma, c_sky, c_txp. Bold shows best result per data set. The first row shows results of the best individual subsystem.

As a general trend we see that the performance increases with the number of classifiers; however, there exist certain subsets which perform better than using all available classifiers.

Best Subset Selection. In Figure 1, we marked for each number of subsystems the highest OOB-F_1-Score on the Dev set by a diamond. In addition, the subset with the overall highest OOB-F_1-Score, consisting of 7 classifiers, is displayed as a filled diamond.

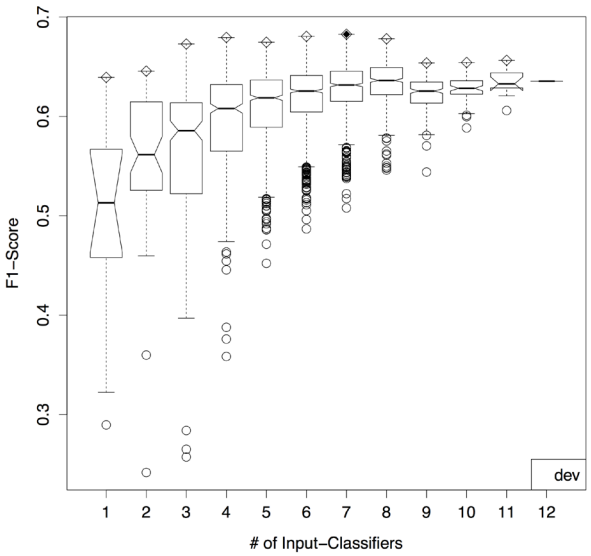

Figure 1: Box Plot showing the F_1 scores (out-of-bag) for all subsets on the Dev set. Diamonds mark the best combination of classifiers for the corresponding number.

We also evaluated the performance of these "best" subsets on other unseen test data. In Figure 2, we show the results of the test set Twitter2014. The scores for the very subsets marked in Figure 1 are displayed in the same way here.

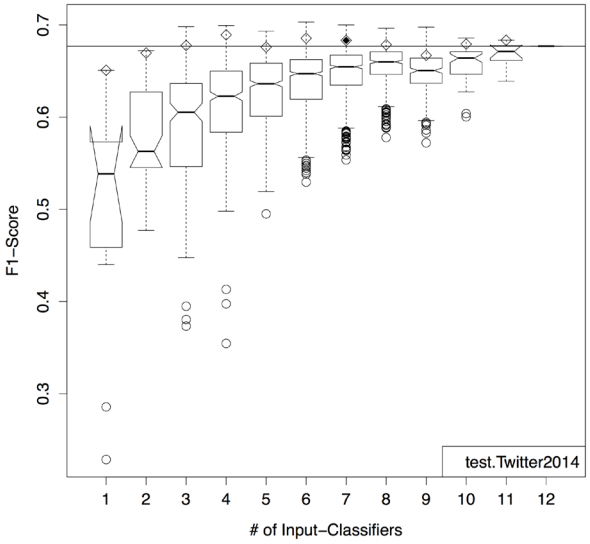

Figure 2: F_1 scores of all subsets on the Twitter2014 test set.

For comparison, we marked the performance of the system with all classifiers by a straight line. We find that all subsets that are "best" on the Dev set perform very well on the Twitter2014 set. In fact, some even beat the system with all classifiers. Similar behaviour can be observed for Twitter2013 and LiveJournal2014 (data not shown), while All_Subsets yields significantly superior results on SMS2013 (see Figure 3). No conclusive observation is possible for Sarcasm2014 (data not shown).

To elucidate on the question whether to use a subset with the highest OOB-F_1 on the Dev set (called Max_OOB_Subset) or to use all available classifiers, we show in Table 3 the performance of these systems on all test sets in rows 2 and 5, respectively. Since All_Systems is in 2 out of 5 cases the best classifier, and "Max_OOB_Subset" in 3 out of 5 cases, a decisive answer cannot be drawn. However, we find

that All_Systems generalizes better to foreign types of data, while Max_OOB_Subset performs well on similar data (in this case, tweets).

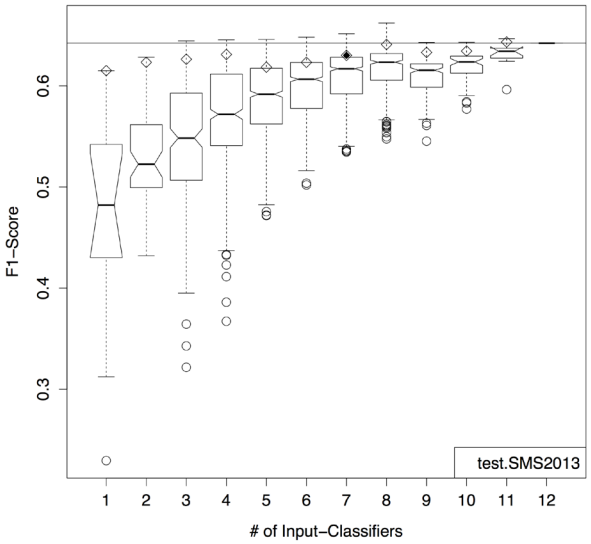

Figure 3: F_1 score of all subsets on the SMS2013 test set.

4 Conclusion

We have shown that a meta-classifier approach using random forest can beat the performance of the individual sentiment classifiers it is based on. Typically, the more subsystems are used, the better the performance. However, there exist selections of only few subsystems that perform comparable to using all subsystems. In fact, a good selection strategy is to select the subset which has maximum out-of-bag F_1 score on the training data. This subset performs slightly better than All_Systems on similar data sets, and only slightly worse on new types of data. Advantage of this subset is that it requires less classifiers (7 instead of 12 in our case), which reduces the cost (runtime or license fees) of the meta-classifier.

5 Acknowledgements

We would like to thank all system providers for giving us the opportunity to use their systems for this evaluation, and especially Tobias Günther and Martin Jaggi for carefully reading the manuscript.

References

Leo Breiman. 2001. Random Forests. Machine Learning 45(1), 5-32.

Mark Cieliebak, Oliver Dürr, Fatih Uzdilli. 2014. Meta-Classifiers Easily Improve Commercial Sentiment Detection Tools. In *Proceedings of the 9th edition of the Language Resources and Evaluation Conference* (LREC), pages 3100-3104, May 26-31, 2014, Reykjavik, Iceland.

Pedro P. Balage Filho, Thiago A. S. Pardo. 2013. NILC USP: A Hybrid System for Sentiment Analysis in Twitter Messages. In *Proceedings of the International Workshop on Semantic Evaluation* (SemEval-2013), pages 568-572, June 14-15, 2013, Atlanta, Georgia, USA.

Gizem Gezici, Rahim Dehkharghani, Berrin Yanikoglu, Dilek Tapucu, Yucel Saygin. 2013. SU-Sentilab: A Classification System for Sentiment Analysis in Twitter. In *Proceedings of the International Workshop on Semantic Evaluation* (SemEval-2013), pages 471-477, June 14-15, 2013, Atlanta, Georgia, USA.

Tobias Günther, Lenz Furrer. 2013. GU-MLT-LT: Sentiment Analysis of Short Messages using Linguistic Features and Stochastic Gradient Descent. In *Proceedings of the International Workshop on Semantic Evaluation* (SemEval-2013), pages 328-332, June 14-15, 2013, Atlanta, Georgia, USA.

Martin Jaggi, Fatih Uzdilli, and Mark Cieliebak. 2014. Swiss-Chocolate: Sentiment Detection using Sparse SVMs and Part-Of-Speech n-Grams. In *Proceedings of the International Workshop on Semantic Evaluation* (SemEval-2014), August 23-24, 2014, Dublin, Ireland.

Morgane Marchand, Alexandru Ginsca, Romaric Besançon, Olivier Mesnard. 2013. [LVIC-LIMSI]: Using Syntactic Features and Multi-polarity Words for Sentiment Analysis in Twitter. In *Proceedings of the International Workshop on Semantic Evaluation* (SemEval-2013), pages 418-424, June 14-15, 2013, Atlanta, Georgia, USA.

Sara Rosenthal, Preslav Nakov, Alan Ritter, and Veselin Stoyanov. 2014. SemEval-2014 Task 9: Sentiment Analysis in Twitter. In *Proceedings of the International Workshop on Semantic Evaluation* (SemEval-2014), August 23-24, 2014, Dublin, Ireland.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, Theresa Wilson. 2013. SemEval-2013 Task 2: Sentiment Analysis in Twitter. In *Proceedings of the International Workshop on Semantic Evaluation* (SemEval-2013), pages 312-320, June 14-15, 2013, Atlanta, Georgia, USA.