IIP at SemEval-2016 Task 4: Prioritizing Classes in Ensemble Classification for Sentiment Analysis of Tweets

Jasper Friedrichs
Infosys Limited
7707 Gateway Boulevard
Newark, CA 94560, USA
jasper_friedrichs@infosys.com

Abstract

This paper describes the submission of team IIP in SemEval-2016 Task 4 Subtask A. The presented system is a novel weighted sum ensemble approach for sentiment analysis of short informal texts. The ensemble combines member classifiers that output classification confidence metrics. For the ensemble classification decision the members are combined by weights. In the presented approach the weights are derived to prioritize specific classes in multi-class classification. The presented results confirm that this improves results for the prioritized classes. The official task submission achieved a macro-averaged negative positive F1 of 57.4%. Post submission changes resulted in a F1 score of 60.2%. The evaluation also shows that the system outperforms other ensemble methods.

1 Introduction

The SemEval workshops offer the opportunity to compete across a variety of natural language processing tasks. The SemEval-2016 Task 4 Subtask A targets message polarity classification of tweets (Nakov et al., 2016). The polarity can be negative, neutral or positive while the submissions are ranked omitting performance on the neutral class. In practical use cases some classes of a multi-classification problem might be deemed more important than others. For example some work looks explicitly at negative sentiment (Tetlock, 2007).

Combining diverse methods has shown success in sentiment analysis. The combination of machine learners and opinion lexicons has resulted in some of the best submissions in previous SemEval competitions (Kiritchenko et al., 2014; Miura et al., 2014). Along the line of combining different methods, ensemble approaches have also shown top results in previous runs of this task. Both ensembles of a small number of sophisticated systems (Hagen et al., 2015) as well as large numbers of simpler approaches have been evaluated (Wicentowski, 2015). Ensemble classification with regard to combining different machine learners and feature spaces has previously been evaluated extensively for document level sentiment classification (Xia et al., 2011). In that context, weighted sum ensemble methods have shown the best performance.

This paper describes a weighted sum ensemble that prioritizes some classes in multi-class classification. Results compare the system against two baselines. One baseline is the equivalent approach without prioritizing classes, while the other is an unweighted combination of ensemble members. Naive Bayes and logistic regression classifiers are explored as members across a variety of feature spaces. These classifiers are know to perform differently (Ng and Jordan, 2002). The presented results show:

1. The presented approach successfully prioritizes classes in a multi-class classification problem.
2. The ensemble method outperforms individual members and the baseline ensembles.

The system description will start by a brief outline of the evaluation data. Then the ensemble members are described before the ensemble method is detailed. Finally, the results on all SemEval test sets allow an assessment of the approach and future work.

Proceedings of SemEval-2016, pages 230–234,
San Diego, California, June 16-17, 2016.

Twitter Corpus	Pos.	Neg.	Neu.	Total
2013-train (A)	2869	1077	3733	7679
2013-dev (A,C)	459	258	569	1286
2013-test	1571	601	1637	3813
2014-test	978	200	668	1853
2015-test	1038	365	987	2392
2016-train (A)	2483	678	1625	4796
2016-dev (A,B,C)	669	319	611	1595
2016-devtest (A,B,C)	786	254	548	1588
2016-test	7060	3231	10342	20633

Table 1: SemEval data subsets as well as the full 2016 training set (A), submission (B) and post-submission (C) development data.

2 Data

Training data for this approach is constrained to data provided through the SemEval competitions. Table 1 shows the evaluation data used in the approach. This is a subset of the original data, as some tweets were unavailable when querying the Twitter API. The test data corresponds to the data used in the official task ranking (Nakov et al., 2016).

3 Ensemble Members

The ensemble members are the basic exchangeable building blocks of this approach. In this work Naive Bayes and logistic regression are chosen as differently performing members.

3.1 Naive Bayes

The Naive Bayes classifier is based on the Bayes theorem. The assumption that features are statistically independent might seem too naive. However, this approach often performs surprisingly well. The implementation uses the multinomial Naive Bayes classifier of the datumbox library[1].

3.2 Logistic Regression and Opinion Lexicons

Logistic regression is the second classification approach for ensemble members. Input for this method are text features, as well as scores from five opinion lexicons. Three lexicons have been created automatically from large corpora, namely SentiWordNet (Baccianella et al., 2010), NRC Hashtag Sentiment Lexicon and Sentiment140 Lexicon (Kiritchenko et al., 2014). Bing Liu's Opinion Lexicon (Hu and Liu, 2004) was created manually and a fifth lexicon was created automatically and then curated manually. For each lexicon, the two sums over all negative as well as positive opinion scores corresponding to unigrams or lemmas in a message are added as features. The implementation uses the logistic regression classifier available in LIBLINEAR (Fan et al., 2008).

3.3 Feature Extraction

User names, URLs and retweet handles are removed before feature extraction. Part-of-speech tags of the CMU ARK Tagger (Owoputi et al., 2013) are used for truecasing if words in a tweet are mostly lowercase or mostly capitalized. ClearNLP (now NLP4J[2]) is used for tokenization, part-of-speech tagging and dependency parsing (Choi and Mccallum, 2013; Choi, 2016). Feature inputs for the classifiers are bag-of-words of unigrams (*uni*), part-of-speech tags (*pos*), bigrams (*bi*), dependency pairs (*dp*) (Xia et al., 2011) and brown clusters (*cl*) (Owoputi et al., 2013). In this work a classifier only ever uses one of the different text feature sets.

4 Ensemble

Ensemble methods combine a set of multiple member classifiers. These members can be various classifiers of different methods and different feature sets. Individual classifiers output a classification decision or a classification score for each class.

In the context of this approach classification scores are required for all ensemble members. These scores $o_{\text{ki}} \geq 0$ for every class $i \in C$ and every member classifier $k \in M$ are assumed to be normalized,

$$\sum_{i=1}^{C} o_{\text{ki}} = 1. \tag{1}$$

The score o_{ki} can be interpreted as a probability or confidence measure of classifier k for class i.

A basic score based classification method would be to derive the classification decision from the sum over all scores. The highest accumulated class sum would determine the decision, as in

$$\arg\max_{i \in C} \sum_{k=1}^{M} o_{\text{ki}}. \tag{2}$$

[1] https://github.com/datumbox/datumbox-framework

[2] https://github.com/emorynlp/nlp4j

Instead the approach uses a weighted sum, with a weight w_{ki} for every classifier k and every class i. Differentiating weights by class represents a finer grained weighting scheme than only differentiating by classifier. The ensemble output is determined by

$$\arg\max_{i \in C} \sum_{k=1}^{M} w_{\mathrm{ki}} o_{\mathrm{ki}}, \tag{3}$$

as the class with the highest weighted sum. The essential aspect of a weighed ensemble approach then is how the weights are calculated. The following sections describe optimization conditions that can be used to calculate weights.

4.1 Standard Weight Optimization

A weighted sum ensemble attempts to improve classification by weighing the class scores of individual members. Weights can be calculated based on a gold dataset where the class scores o_{ki} and the correct gold label g are known.

The decision function (3) is based on the maximal weighted sum of scores. It is straight forward that a lower difference between weighted sums of different classes is more prone to an erroneous decision through inaccuracies. Thus an intuitive condition for optimal weights could be to maximize the difference between the weighted sum for the correct class and all sums of incorrect classes. For every known gold label g and the corresponding scores o_{kg} the conditions would be

$$\sum_{k=1}^{M} w_{\mathrm{kg}} o_{\mathrm{kg}} - w_{\mathrm{kj}} o_{\mathrm{kj}} = |M|, \tag{4}$$

for all labels besides the gold label, $j \in C \setminus \{g\}$. The unweighted sum of classification scores was defined equal one for a single classifier (1). This would also be the maximal difference in case of one classifier. Consequently, the conditions for maximal difference between weighted sum scores over the classifier set M are set equal to the cardinality of the set.

4.2 Prioritizing Weight Optimization

In contrast to the previously introduced weight optimization conditions the following conditions aim to prioritize valid classification of some classes over others. Low priority classes are defined by the set $L \subseteq C$. This also defines the priority classes as $P = C \setminus L$.

The approach does not aim to improve the ensemble classification across low priority classes L. For a low priority label $l \in L$ the weights are fixed to

$$w_{\mathrm{kl}} = 1. \tag{5}$$

Based on this, the standard weight conditions (4) for any low priority gold label $g \in L$ and all priority labels $p \in P$ are rephrased as

$$\sum_{k=1}^{M} w_{\mathrm{kp}} o_{\mathrm{kp}} = -|M| + \sum_{k=1}^{M} o_{\mathrm{kg}} \leq 0. \tag{6}$$

This is problematic because the unweighted sum over scores is positive by definition, since scores can't be negative. However, the derivation shows that the priority weights w_{kp} would be conditioned to change this sum to negative in favor of low-priority classification decisions. This is a contradiction to the concept of priority classes. The conditions (6) for any low priority gold label $g \in L$ and all priority labels $p \in P$ are relaxed to

$$\sum_{k=1}^{M} w_{\mathrm{kp}} o_{\mathrm{kp}} = 0. \tag{7}$$

This can be understood as a lower bound for priority weights. Priority weights are also still conditioned to improve priority classification decisions as per the standard conditions (4) for any $g \in P$.

Based on the gold dataset the conditions for low priority gold labels (7) and for priority gold labels (4) form an overdetermined system of equations. The solution to this are the priority weights optimized to improve the decision of the ensemble approach. The weights are calculated by solving the conditions as a least squares problem. This requires gold labels from a development dataset different from classifier training data.

5 Results

The following section compares results of the introduced approach against the ensemble members and two baseline ensemble methods. The results for Naive Bayes and logistic regression ensemble members on different feature spaces as well as the results for the ensembles are presented in the following.

Test, weight data	Sum			IIP std			IIP prio		
	uni-bi	uni-bi-cl	all	uni-bi	uni-bi-cl	all	uni-bi	uni-bi-cl	all
2013-test, B	55.5	56.8	55.7	55.1	55.4	53.7	**60.5**	**60.5**	59.8
2014-test, B	61.6	62.9	61.3	59.5	60.5	59.4	65.3	**65.9**	64.5
2015-test, B	57.4	59.4	58.3	58.5	57.6	56.8	**63.3**	62.9	61.0
2016-test, B	56.1	58.0	56.6	55.0	56.0	55.4	58.3	**58.7**	57.4∗
2013-test, C	55.5	56.8	55.7	59.7	59.4	59.7	63.6	64.3	**64.6**
2014-test, C	61.6	62.9	61.3	64.7	66.1	65.3	68.0	**69.9**	68.5
2015-test, C	57.4	59.4	58.3	60.7	62.3	61.3	64.1	**66.2**	65.5
2016-test, C	56.1	58.0	56.6	58.2	59.0	58.8	59.4	**60.2**	59.5

Table 2: Macro-averaged positive negative F1 [%] for all test data sets across three ensemble methods and three member sets. Set *all* corresponds to all classifier, feature combinations evaluated for 2016-test data in Table 3. Ensemble members were trained on the full 2016 training set (A) while ensemble weights were optimized on 2016 (B) and 2013/2016 development sets (C, Table 1).

2016-test	uni	pos	bi	dp	cl
NB	54.5	30.3	40.7	37.6	**54.4**
LR	55.5	53.5	52.0	52.7	**57.6**

Table 3: Macro-averaged positive negative F1 [%] for ensemble members of 2016-test on full 2016 training set (A, Table 1).

Table 3 shows the results for Naive Bayes (*NB*) and logistic regression (*LR*) ensemble members. For both classification approaches brown clusters (*cl*) show the best performance.

Table 2 shows results for three ensemble approaches and three sets of members. The *Sum* columns show results for an unweighted sum over contributing classifier scores, as in (2). *IIP std* is a weighted sum approach with standard weight optimization as in (4). *IIP prio* adds the priority condition (7) with positive and negative as priority classes.

The bottom result set for weight optimization on 2013 and 2016 development data shows substantially better results than the top one, where weights were optimized on 2016 development data. While the weighted sum approach is of course unaffected by this, this holds true for all ensemble member sets in the weighted sum ensembles.

Across both result sets the *IIP prio* ensemble always outperforms the other two baseline ensemble methods. The standard ensemble which does not prioritize classes *IIP std* outperforms the sum baseline in the bottom result set but often does not in the top one. For all ensemble approaches the member sets of classifiers for unigram, bigram and cluster feature spaces usually show the best results.

The system for the official submission ∗ used all members with priority weight optimization, obtaining a macro-averaged F1 of 57.4%. Though this outperforms the equivalent baseline ensembles it performs on a similar level as the best logistic regression member in Table 3. In contrast the best *IIP prio* result used unigram, bigram and cluster members in an ensemble optimized for 2013 and 2016 development data, achieving a macro-averaged F1 of 60.2%.

6 Conclusion

This paper presented two methods for weighted sum ensemble classification. The introduced class prioritizing method outperformed the standard method in all evaluations. Furthermore, the results show that the class prioritizing weight ensemble method usually outperformed the basic sum ensemble approach substantially. This shows that combining different classifiers across various feature spaces while prioritizing some classes in multi-class classification works well with the presented system.

The results varied significantly depending on the data used for optimizing weights. Optimization on a more diverse data set showed better performance. Questions of domain dependence and over-fitting need to be explored further. With the modular nature of an ensemble a variety of classifiers and features are left to be evaluated in the context of this approach.

Acknowledgments

The author is grateful for the opportunity to contribute the implementation of this approach to NLP4J[3].

[3]https://github.com/emorynlp/nlp4j

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *in Proc. of LREC*.

Jinho D. Choi and Andrew Mccallum. 2013. Transition-based dependency parsing with selectional branching. In *In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*.

Jinho D. Choi. 2016. Dynamic feature induction: The last gist to the state-of-the-art. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (NAACL 2016)*.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Matthias Hagen, Martin Potthast, Michel Bchner, and Benno Stein. 2015. Webis: an ensemble for twitter sentiment detection. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015*, pages 582–589. Association for Computational Linguistics, June.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M. Mohammad. 2014. Sentiment analysis of short informal texts. *J. Artif. Intell. Res. (JAIR)*, 50:723–762.

Yasuhide Miura, Shigeyuki Sakaki, Keigo Hattori, and Tomoko Ohkuma. 2014. Teamx: A sentiment analyzer with enhanced lexicon mapping and weighting scheme for unbalanced data. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 628–632, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Fabrizio Sebastiani. 2016. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval 2016)*, San Diego, California, June. Association for Computational Linguistics.

Andrew Y. Ng and Michael I. Jordan. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 841–848. MIT Press.

Olutobi Owoputi, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A. Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *In Proceedings of NAACL*.

Paul C. Tetlock. 2007. Giving Content to Investor Sentiment: The Role of Media in the Stock Market. *Journal of Finance*, 62(3):1139–1168, 06.

Richard Wicentowski. 2015. Swatcs65: Sentiment classification using an ensemble of class projects. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 631–635, Denver, Colorado, June. Association for Computational Linguistics.

Rui Xia, Chengqing Zong, and Shoushan Li. 2011. Ensemble of feature sets and classification algorithms for sentiment classification. *Inf. Sci.*, 181(6):1138–1152, March.