Swiss-Chocolate: Combining Flipout Regularization and Random Forests with Artificially Built Subsystems to Boost Text-Classification for Sentiment

Fatih Uzdilli
Zurich University of Applied Sciences
Switzerland
uzdi@zhaw.ch

Martin Jaggi
ETH Zurich
Switzerland
jaggi@inf.ethz.ch

Dominic Egger
Zurich University of Applied Sciences
Switzerland
dominicegger@bluewin.ch

Pascal Julmy
Zurich University of Applied Sciences
Switzerland
pascal.julmy@gmail.com

Leon Derczynski
University of Sheffield
UK
leon@dcs.shef.ac.uk

Mark Cieliebak
Zurich University of Applied Sciences
Switzerland
ciel@zhaw.ch

Abstract

We describe a classifier for predicting message-level sentiment of English micro-blog messages from Twitter. This paper describes our submission to the SemEval-2015 competition (Task 10). Our approach is to combine several variants of our previous year's SVM system into one meta-classifier, which was then trained using a random forest. The main idea is that the meta-classifier allows the combination of the strengths and overcome some of the weaknesses of the artificially-built individual classifiers, and adds additional non-linearity. We were also able to improve the linear classifiers by using a new regularization technique we call flipout.

1 Introduction

With the availability of huge amounts of user generated text online, the interest in automatic sentiment analysis of text has greatly increased recently in both academia and industry.

The goal is to classify a tweet (on the full message level) into the three classes positive, negative, and neutral. In this paper, we describe our approach using a modified SVM based classifier on short text as in Twitter messages. Our system has participated in the SemEval-2015 Task 10 competition, "Sentiment Analysis in Twitter, Subtask–B Message Polarity Classification" (Rosenthal et al., 2015). Previous iterations of the evaluation were run in 2013 and 2014.

Our Results in the Competition. Our system was ranked 8th out of 40 participants, with an F1-score of 62.61 on the Twitter-2015 test set. The 2015 winning team obtained an average F1-score of 64.84.

The detailed rankings of our approach were: 4th rank on the LiveJournal data; 6th on the SMS data (2013); 10th on Twitter-2013; 12th on Twitter-2014; and 25th on Twitter Sarcasm. See (Rosenthal et al., 2015) for full details and all results.

Data. In the competition, tweets for training and development were provided as tweet IDs. A fraction (10-15%) of the tweets were no longer available on Twitter, which made results of the competition not fully comparable. For testing, in addition to last year's data (tweets, SMS, LiveJournal), new tweets were provided. An overview of the data that we were able to download is given in Table 1.

Our Approach. Our system is based on two main ideas. First, we propose a new regularization technique called *flipout*, which post-processes a trained classifier model for better generalization performance. Details of this are given in Section 2. Second, we combine multiple classifiers with a meta-classifier, to yield better performance than each single sub-classifier (Dürr et al., 2014; Cieliebak et al., 2014). To achieve this, we extended our existing system (Jaggi et al., 2014). The result is simple: a large collection of features used in a linear SVM classifier. We replicated that system with several dif-

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 608–612,
Denver, Colorado, June 4-5, 2015.

ferent choices of features and parameters. The output of all those artificially built classifiers is then feed as input to a random forest classifier, which generated final classification results, and gave our system additional non-linear output capabilities.

Table 1: Overview of the data we found available for training, development and testing.

Dataset	Total	Posit.	Negat.	Neutr.
Train (Tweets)	8224	3058	1210	3956
Dev (Tweets)	1417	494	286	637
Test: Twitter2015	2390	1038	365	987
Test: Twitter2014	1853	982	202	669
Test: Twitter2013	3813	1572	601	1640
Test: SMS2013	2093	492	394	1207
Test: Tw2014Sarcasm	86	33	40	13
Test: LiveJournal2014	1142	427	304	411

2 Flipout Regularization

We propose a new kind of post-processing/regularization technique to improve classification accuracy in a setting with several different available datasets. The intuition comes from the setting of transfer learning. Many words in the training data do not occur in the same context as in the target data (as for example caused by topic shifts, such as in the evaluation task's scenario here). By finding suitable replacements for some input words, the generalization performance of a pre-trained linear classifier can be improved. Since this post-processing of a pre-trained classifier overrides potentially many of its weights, the post-processing has an additional regularizing effect with respect to the original training set, in addition to the transfer effect towards the target dataset.

We follow a greedy approach to find the best word-replacements which is as follows:

1. Split the dataset into 4 parts, here called fliptrain, flipdev1, flipdev2 and fliptest.
2. Train a classifier (e.g. SVM) on the set fliptrain, using the original full set of features.
3. Calculate prediction score on datasets flipdev1 and flipdev2.
4. Pick a subset S of words from the vocabulary of fliptrain. This is the *word-pool* for the flipout trick.
5. For each word $w_1 \in S$:
   - For each word $w_2 \in S$:
     Consider the modified classifier using the replacement (flip) of input words $w_1 \mapsto w_2$. Compute its prediction score on the validation datasets flipdev1 and flipdev2.
   - Keep the replacement $w_1 \mapsto w_2^*$ which resulted in the maximum improvement for the word w_1, in the sense of min(improvement on flipdev1, improvement on flipdev2),

One would expect that this approach would replace words of the original set (fliptrain) with words having a better discriminative power on the new set (flipdev). In reality, it turned out that words without an obvious relation to each other were replaced such as: *2nd* $\mapsto$ *may*, *about* $\mapsto$ *I'm*, *we* $\mapsto$ *day*, etc. The reason we have separated the development sets (flipdev1 and flipdev2) is to better avoid potential overfitting.

3 System Description

For our system, we preprocessed the tweets and extracted textual features. Using different subsets of these features and flipout, we train different linear classifiers resulting in sentiment classification systems which are intrinsically different from each other. These "subsystems" were then combined into a meta-classifier using a random forest (Breiman, 2001). The random forest 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.

3.1 Preprocessing

The tweets were preprocessed with standard methods before extracting the features.

- URLs and usernames are each normalized to a replacement token
- Tokenizer: We used ArkTweetNLP (Owoputi et al., 2013) which is suitable for tweets. All text was transformed to lowercase (except for special features relying on case information).
- Negation encoding: The negated context of a sentence is marked as in (Pang et al., 2002), us-

ing the list of negation words from Christopher Potts' sentiment tutorial[1].

3.2 Features for the Subsystems

The subsystems use different subsets of the features we introduce here. Most of them are the same as in our last years submission (Jaggi et al., 2014). New additions are marked with a + sign.

Features:

- **n-grams**: presence of word n-grams ($n = 1...4$)
- **POS-n-grams**: presence of word n-grams with one or more words replaced by the POS-Tag (Jaggi et al., 2014). The ArkTweetNLP structured prediction POS tagger provided by (Owoputi et al., 2013) together with their provided standard model (model.20120919) suitable for Twitter data was used ($n = 3...5$)
- **non-contiguous n-gram**: presence of word n-grams with one or more words replaced by a wildcard ($n = 3...5$)
- **character n-grams**: presence of character n-grams ($n = 3...6$) weighted increasingly by their length (weights $0.7 \cdot \{1.0, 1.1, 1.2, 1.4, 1.6, 1.9\}$ for length $3, 4, \ldots$)
- **# upper cased**: number of tokens written with all characters in upper case
- **# of hashtags**
- **# of POS tags**: for each POS-tag the number of occurrences
- **continuous punctuation**: number of continuous exclamation marks, number of continuous question marks (max)
- **last token punctuation**: whether the last token contains an exclamation mark or question mark or a period
- **# elongated words** number of words which repeat the same character more than two times
- **# negated tokens** the number of words occurring in a negation context
- **Lexicons**: For each lexicon (NRC-emotion, BingLiu, MQA, NRC-HashtagSentiment, Sentiment140, Sentiment140-3-class, RottenTomatoes-3-class):

  - total tokens for each class (positive, negative and neutral for 3-class lexicons)
  - score of last token for each class
  - maximum score over all tokens for each class
  - total score over all tokens for each class
  - +score of last token regardless of the class
  - +maximum score over all tokens for all classes together
  - +total score over all tokens

  For the 2-class lexicons, we flip the score of tokens occurring in the negation scope. The 3-class lexicons are already trained with marked negations (Jaggi et al., 2014).
- **+lemma n-grams**: presence of lemma n-grams ($n = 1...4$), by using the Standford Core NLP lemmatizer.
- **+cluster unigram**: whether a word from each cluster in the CMU tweet clusters occurs or not
- **+GloVe**: GloVe word embeddings (Pennington et al., 2014) are a newer version of the word2vec embedding by (Mikolov et al., 2013), using a matrix factorization instead of deep learning. We used the sum, minimum and maximum of the GloVe-vectors for the tokens occuring in the tweet.

3.3 Subsystems

For the subsystems we used different linear classifier variants trained using the LibLinear package (Fan et al., 2008), all being multi-class classifiers for the three classes in a one-against-all setting.

Subsystem 1. We combined all features to a single feature vector using an ℓ_1-regularized squared loss SVM classifer and flipout regularization as described in Section 2. We trained the SVM and optimized the regularization parameter using 10-fold cross-validation on the training set. The remaining sets were used for flipout: dev as flipdev1 and twitter-test13 as flipdev2 and twitter-test14 for testing. We chose the word pool S for flipout as the most frequent 50 words in fliptrain.

Subsystem 2. The same as subsystem 1 but without flipout. The system was trained on train+dev and the SVM regularization parameter C was optimized against the test sets.

[1] http://sentiment.christopherpotts.net/lingstruc.html

Subsystem 3. The same as subsystem 2 but using Logistic Regression instead of SVM.

Subsystem 4. The same as subsystem 2 but without any lexicon features.

Subsystem 5. The same as subsystem 2 but using only the GloVe word-embedding features.

3.4 Meta-Classifier

Each subsystem outputs three real values corresponding to the three sentiment classes. In addition, it outputs the categorical value for the predicted sentiment class. Our meta-classifier used these 4 values as input features. We trained a random forest using the Weka Java-library on the train data, although the subsystems are trained on the same data. To avoid overfitting, we regularized the random forest against the test sets by trying different values for number of trees, maximum depth of the forest and the number of features used per random selection.

4 Results

	Subsystem 1	Subsystem 2	Subsystem 3	Subsystem 4	Subsystem 5	**Final System**
Twitter15	62.70	62.07	62.41	53.72	58.11	62.73
Twitter14	69.44	69.07	69.34	61.60	63.42	70.19
Twitter13	69.64	69.05	69.49	61.73	61.84	69.70
LiveJournal14	73.54	74.14	74.29	62.32	62.67	74.48
Sarcarsm14	52.94	52.15	50.69	56.15	56.17	49.83

Table 2: Results of our subsystems and final system.

Overall Performance. Looking at the overall performance, we managed to increase the scores on every test set compared to our previous year's submission. Table 2 shows the scores of our individual subsystems as well as the final system on each test set. Note that our results in the official submission are slightly different from Table 2, because of a mistake we made in the class assignments in our random forest input, which is fixed here.

Classifiers. Subsystems 2 and 3 only differ in the choice of the linear classifier. Our results here show that logistic regression slightly outperforms SVM.

Flipout Regularization. Flipout proved very useful. Subsystem 1 (with flipout) reached from 0.37 to 0.79 higher F1 than Subsystem 2 (without flipout).

Features from Unsupervised Learning: Lexicons and Word-Embeddings. Subsystem 4 does not use any of the lexicon features which were constructed on a separate unlabeled large corpus. The large decrease in performance shows the importance of the lexicons. Also we can see that Subsystem 5 (which only uses the GloVe word-embedding features) results in a very small variation of its scores on the different test sets, compared to the other systems. This confirms our expectation that features generated from unsupervised training on a large data set will generalize better, i.e. are more robust to topic and domain changes.

Meta-Classifier. The final system compared with Subsystem 1 shows the gain from performing meta-classification. On last year's Twitter test set, we obtain an improvement of 0.75 F1-Score. On this year's test set (which was hidden), we achieved an improvement of 0.03, which was low. However, we are encouraged by the large improvement of 0.94 F1-Score on out-of-domain data (Live Journal) – which was not seen during training.

5 Conclusion

We have described a classifier to predict the sentiment of short texts such as tweets. Our system is built upon the approach of our previous systems (Jaggi et al., 2014) and (Dürr et al., 2014), with several modifications and extensions in features and regularization. We have seen that our system significantly improves upon last year's approach, achieving a gain of 2.65 points in F1 score on last year's test data.

We showed that our newly introduced flipout regularization technique improved the score on our system. To be able to make general statements we need to further investigate its behavior on different data sets. We also showed that artificially-built subsystems can be used to improve upon the best classifier using meta-classification. A question which remains is how one could automatize the meta-classification approach to built the most beneficial subsystems.

References

Leo Breiman. (2001). Random Forests. *Machine Learning*, 45(1):5–32.

Mark Cieliebak, Oliver Dürr, and Fatih Uzdilli (2014). Meta-Classifiers Easily Improve Commercial Sentiment Detection Tools. In *LREC 2014 - Proceedings of the 9th International Conference on Language Resources and Evaluation*.

Oliver Dürr, Fatih Uzdilli, and Mark Cieliebak (2014). JOINT FORCES: Unite Competing Sentiment Classifiers with Random Forest. In *SemEval 2014 - Proceedings of the 8th International Workshop on Semantic Evaluation*, pages 366–369, Dublin, Ireland.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin (2008). LIBLINEAR: A Library for Large Linear Classification. *JMLR*, 9:1871–1874.

Martin Jaggi, Fatih Uzdilli, and Mark Cieliebak (2014). Swiss-Chocolate: Sentiment Detection using Sparse SVMs and Part-Of-Speech n-Grams. In *SemEval 2014 - Proceedings of the 8th International Workshop on Semantic Evaluation*, pages 601–604, Dublin, Ireland.

Tomas Mikolov, Quoc V Le, and Ilya Sutskever (2013). Exploiting Similarities among Languages for Machine Translation. *arXiv.org*.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith (2013). Improved Part-Of-Speech Tagging for Online Conversational Text with Word Clusters. In *Proceedings of NAACL-HLT*, pages 380–390.

Bo Pang, Lillian Lee and Shivakumar Vaithyanathan (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. In *ACL 2002 - Conference of the Association for Computational Linguistics*, pages 79–86, Morristown, NJ, USA.

Jeffrey Pennington, Richard Socher and Christopher D Manning (2014). GloVe: Global Vectors for Word Representation, EMNLP 2014 - Conference on Empirical Methods in Natural Language Processing.

Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M Mohammad, Alan Ritter and Veselin Stoyanov (2015). SemEval-2015 Task 10: Sentiment Analysis in Twitter. In *Proceedings of the 9th International Workshop on Semantic Evaluation*, Denver, Colorado.