Classifier Calibration for Multi-Domain Sentiment Classification

Stephan Raaijmakers and **Wessel Kraaij**
TNO ICT, Delft, The Netherlands

Abstract

Textual sentiment classifiers classify texts into a fixed number of affective classes, such as positive, negative or neutral sentiment, or subjective versus objective information. It has been observed that sentiment classifiers suffer from a lack of generalization capability: a classifier trained on a certain domain generally performs worse on data from another domain. This phenomenon has been attributed to domain-specific affective vocabulary. In this paper[1], we propose a voting-based thresholding approach, which calibrates a number of existing single-domain classifiers with respect to sentiment data from a new domain. The approach presupposes only a small amount of annotated data from the new domain. We evaluate three criteria for estimating thresholds, and discuss the ramifications of these criteria for the trade-off between classifier performance and manual annotation effort.

Introduction

It is widely known (e.g. Aue and Gamon (2005); Andreevskaia and Bergler (2008)) that sentiment classifiers suffer from domain specificity: while they may perform well for the domain they are trained on, their performance usually drops when applied to other domains. This is not surprising, as sentiment is often subtly expressed with domain-specific vocabulary. Yet, it seems intuitive that a large portion of the sentimental vocabulary is shared across domains: affective terms like *awkward, bad, good, terrible, terrific* are universally applicable to all imaginable domains, with urban exceptions like 'that's a bad car' in a positive sense. If textual domains do indeed show overlap of sentiment vocabulary, then it would make sense to attempt to identify these patterns of shared expression, and combine them into aggregate data sets. In this paper, we propose a calibration technique that can be used for thresholding a classifier, and which factors out hard cases that cannot be reliably assigned to one of the classes the classifier is trained on. This procedure additionally generates a trade-off between classifier performance and manual inspection. Our method compares favorably with previous work, and has low computational complexity.

[1]This research was supported by the PetaMedia Network of Excellence and has received funding from the European Commission's 7th Framework Program under grant agreement no. 216444.

Related work

Applying classifiers to 'out-of-domain' test data is a problem known as the domain-transfer problem (e.g. Ben-David et al. (2009); Wu et al. (2009)). Li and Zong (2008) put forward a classifier fusion approach, comparing feature fusion with classifier combination. The latter option significantly outperforms single-domain classification. Their approach is akin in aim and philosophy to ours, with the difference that our approach aims to alleviate an engineer's workflow by factoring out hard cases and exempting them from classification. Blitzer et al. (2007) use a structural correspondence learning (SCL) model for sentiment classification. SCL identifies correspondences between features from different domains by finding correlations of these features domain-independent pivot features, such as features addressing common frequency among domains, and mutual information with class labels. Depending on the domain, a small number (50) of labeled examples allows their model to adapt itself to a new domain. However, performance and the minimum number of labeled in-domain examples was found to depend on the similarity between the old and new domains. Li et al. (2009) use constrained non-negative matrix factorizations of term-document matrices in order to transfer knowledge from one sentiment domain to another. Andreevskaia and Bergler (2008) advocate the use of WordNet features for cross-domain sentiment classification, in combination with a precision-based voting strategy.

Calibration technique

In our experiments, we deployed Support Vector Machines with the Negative Geodesic Kernel proposed by Zhang et al. (2005) (see also Raaijmakers (2009) and Raaijmakers and Kraaij (2009))[2]. Kraaij et al. (2008) have proposed a thresholding technique for classifier calibration that optimizes a classifier for a pre-specified minimum accuracy. This mechanism allows for balancing the trade-off between classifier performance and manual effort. In order to calibrate our SVMs, we used a similar two-threshold estimation technique. The algorithm estimates two thresholds on the raw output of the SVM decision function: a lower threshold (θ_l) below which data points belong to the negative class,

[2]We implemented this kernel using LIBSVM; see Chang and Lin (2001).

and an upper threshold (θ_u) above which data points belong to the positive class. Data for which the SVM produces a value in between the two thresholds are considered to be 'out of domain' (OOD): these are the hard cases the classifier cannot reliably allocate to one of the two classes. It is defined as

$$OOD = \mid \{d_i \in D \mid \theta_l < C(d_i) < \theta_u\} \mid \quad (1)$$

The OOD class of observations typically could be handed over to an analyst for manual inspection in the case of labeled data, or manual annotation in the case of unannotated test data. The threshold estimation algorithm optimizes a performance function f defined on the quantities TP (true positives), FP (false positives), TN (true negatives), FN (false negatives), OOD; f can be either accuracy, F-score or *yield*:

$$yield = \frac{TP + FP + TN + FN}{TP + FP + TN + FN + OOD} \quad (2)$$

The yield of a classifier is the proportion of data that is not classified as OOD. Optimizing for accuracy will possibly generate a relatively large proportion of OOD data, whereas this proportion will be lower when optimizing for F-score or yield, which, in turn, will lead to lower accuracies. This trade-off between accuracy (or more general: classifier performance) and the amount of data that cannot be analyzed automatically with high accuracy has been closely investigated in a wide array of experiments, described in the next Section. We used a stepsize of 0.01 for thresholding in all experiments.

Experiments and results

Our data consists of the multi-domain sentiment dataset provided by Blitzer et al. (2007). This dataset[3] contains sentiment review data (polarity) from Amazon.com for four different product domains: books, dvd, electronics and kitchen (2,000 reviews each). We used the balanced, preprocessed version that was also used by Blitzer et al. (2007), where each data set contains exactly 1,000 positive and 1,000 negative reviews. Features consist of L1-normalized unigram and bigram frequencies. In our test data sets, the balanced class distribution was preserved (average 50/50), which is why we report accuracy. Another reason is that we compare our results to Blitzer et al. (2007)), who report accuracy as well. We performed three types of experiments with this data. First, we assessed the performance of uncalibrated classification, both within domains (training and test data are taken from the same domain) and across domains (training data is taken from a different domain than test data). These results set a baseline for our subsequent experiments, which fall into two categories: in-domain thresholding and cross-domain thresholding. For in-domain thresholding, we estimate thresholds on development data from a certain domain, train on the corresponding training data, and test on the corresponding test data. After the test data has been classified, we threshold the raw decision values produced by the classifier and allocate the test data points to three classes: the positive class, the negative class, and an 'out of domain' class. For our uncalibrated classification experiments, we applied the same splitting strategy as Blitzer et al. (2007) and divided the four data sets into a training portion of 1,600 reviews, and a test part of 400 reviews. For the calibrated experiments, that depend on development data for threshold estimation, we split the 1,600 data points of the training sets into 1,000 (training) and 600 (development data). The latter data sets were split into a development training set (500) and a development test set (100). On the latter two types of data sets, the thresholds for the calibrated classifiers were estimated: training took place on the development training part, after which the trained classifier was applied to the development test set. Subsequently, the output decision values for the development test data set were thresholded w.r.t. the ground truth of that test data set in order to optimize either accuracy, F-score or yield. Effectively only 100 data points were used for calibration. For cross-domain calibration, the procedure was as follows. Given two domains, development test data was produced as just described. Subsequently, the decision values produced for both test sets after separate classification were combined and thresholded, using the combined two development test sets as ground truth. After thresholded classification, we measured both accuracy and the percentage of data points that are out of domain ('OOD'). A good classifier clearly minimizes OOD and maximizes accuracy. In order to derive an aggregate, unbiased performance measure ('P'), we use the harmonic mean of the percentage of 'in-domain' data points ('ID') and accuracy:

$$\begin{aligned} ID &= 1 - \frac{OOD}{TP+FP+TN+FN+OOD} \\ P &= \frac{2 \times ID \times Accuracy}{ID + Accuracy} \end{aligned} \quad (3)$$

Maximum values of P are obtained only when both ID and accuracy are high. The measure assigns equal weight to ID and accuracy, and reaches 100 only when both quantities are 100. The P-measure can be used as a criterion for selecting a domain that adapts well to a new target domain. Special weighted versions can be devised that penalize certain quantities, such as false positives, or the amount of OOD. This relates to so-called intrinsic (task-specific) cost measures (e.g. Teufel (2007)). Tables 1 and 2 contain accuracy results obtained using uncalibrated classifiers: either in-domain (e.g. training on dvd, and testing on dvd) or cross-domain (e.g. training on dvd and testing on books). In addition, we merged the training data of all four domains, and tested on the four test sets. The in- and cross-domain results in Table 1 show that, in a number of cases, a classifier trained on a different domain than it is tested on is competitive with a classifier trained and tested on the same domain data[4]. For instance, dvd reviews appear akin to book reviews, and kitchen reviews to electronics. This relation apparently is

[3] See `http://www.cs.jhu.edu/~mdredze/datasets/sentiment`.

[4] Following Blitzer et al. (2007) we use the notation $A \rightarrow B$ for training on the training data of domain A and testing on the test data of domain B.

	Acc	SCL	SCL-50
books→ books	78.8	80.4	80.4
dvd→ books	78.8	79.7	-
electronics→ books	69.5	75.4	76
kitchen→ books	76.3	70.9	73.2
Average books	75.9	76.6	
books→ dvd	76.8	77.2	78.5
dvd→ dvd	82.3	82.4	82.4
electronics→ dvd	71	76.2	-
kitchen→ dvd	77.3	76.9	76.6
Average dvd	76.9	78.2	
books→ electronics	68.3	77.5	76.6
dvd→ electronics	71.8	74.1	77.9
electronics→ electronics	86.5	84.4	84.4
kitchen→ electronics	83.3	86.8	-
Average electronics	77.5	80.7	
books→ kitchen	70	78.9	80.7
dvd→ kitchen	75	81.4	-
electronics→ kitchen	82.3	85.9	85.9
kitchen→ kitchen	88.8	87.7	87.7
Average kitchen	78.7	83.5	

Table 1: Accuracy results for in- and cross-domain classification. The SCL columns contain the maximum results obtained by Blitzer et al. (2007), either without (SCL) or with (SCL-50) addition of 50 labeled target domain instances for a selected number of domains.

	Accuracy
Leave-domain-out	
all minus books→ books	79.5
all minus dvd → dvd	81.3
all minus electronics → electronics	82.3
all minus kitchen → kitchen	87.3
Use-all	
all → books	82
all → dvd	83
all → electronics	87.3
all → kitchen	88.8

Table 2: Multi-domain classification.

not symmetric: training a classifier on dvd reviews and applying it to book reviews yields better results than vice versa. This phenomenon was also observed by Blitzer et al. (2007). Yet, the overall picture is that data from different domains may be useful for classifying another domain. This underlines our intuition that the sentiment vocabulary, while being to some extent domain dependent, is to a much larger extent domain independent. Table 2 shows that it is beneficiary to combine different data sources. We tested two conditions: a 'leave-domain-out' condition where we used all domains except the domain the test data belonged to, and a 'use-all' condition where we used the training data for all four domains to classify a particular test set. The results show that in all cases, using all data improves on leaving out the test domain data, and that in all cases, except for kitchen, using all data improves on a single-source, domain-specific classifier (Table 1). We investigated the effect of calibration on single-source, in-domain classifiers (e.g. training on books and testing on books). Results are in Table 3. Optimizing for accuracy leads to relatively large proportions of OOD for these in-domain classifiers, separating hard cases from easy cases. Comparing these results to the in-domain classification results in Table 1, we notice performance gains: results are higher for books (accuracy rises from 78.8 to 83 for thresholding with F and a low OOD), dvd (82.3 rises to 83). For electronics and kitchen, performance is maintained.

Dataset	Thresholding	% OOD	Accuracy	P
books	accuracy	42.3	88.5	69.9
	F	0.8	83	90.4
	yield	0.03	80.3	89.1
dvd	accuracy	46.3	92.5	68
	F	0.08	82.8	90.6
	yield	0.13	83	90.7
electronics	accuracy	48	93.8	66.9
	F	0.08	86.8	92.9
	yield	0.08	86.8	92.9
kitchen	accuracy	48	94	67
	F	0.05	89	94.2
	yield	0	88.8	94.1

Table 3: In-domain thresholding.

θ_{Acc}	% OOD	Acc	P
d→ b	50.5	86.8	63.1
e→ b	41.8	78	66.7
k→ b	52.8	84	60.4
Average b	48.4	82.9	63.6
b→ d	37.5	84.8	72
e→ d	43	83.5	67.8
k→ d	43.5	85	67.9
Average d	41.3	84.4	69.2
b→ e	33.3	76.8	71.4
d→ e	36.5	78.8	70.3
k→ e	44.5	88.5	68.2
Average e	38.1	81.4	70.3
b→ k	52.8	77	58.5
d→ k	46	86.3	66.4
e→ k	49.5	92	65.2
Average k	49.4	85.1	63.5

θ_F	% OOD	Acc	P
d→ b	1.5	73.5	84.2
e→ b	26.5	76.3	74.9
k→ b	29.5	81.5	75.6
Average b	19.2	77.1	78.9
b→ d	1.8	79.8	88
e→ d	9.3	75	82.1
k→ d	9	79.3	84.8
Average d	6.7	78	85
b→ e	6.8	72.8	81.8
d→ e	6.3	73.5	82.4
k→ e	0.8	82.8	90.3
Average e	4.6	76.4	84.9
b→ k	29.5	66	68.2
d→ k	7.8	80.3	85.8
e→ k	0.5	83.8	91
Average k	12.6	76.7	81.7

θ_{yield}	% OOD	Acc	P
d→ b	1.5	73.5	84.2
e→ b	1	70	82
k→ b	1.5	71.5	82.9
Average b	1.3	71.7	83.1
b→ d	1.3	75.5	85.6
e→ d	0.8	71	82.8
k→ d	1.3	75	85.2
Average d	1.1	73.8	84.5
b→ e	1.3	70	81.9
d→ e	1.3	71.5	82.9
k→ e	1	81.5	89.4
Average e	1.2	74.3	84.8
b→ k	1.3	75.5	85.6
d→ k	1.5	77	86.4
e→ k	0.8	83.5	90.7
Average k	1.2	78.7	87.6

Table 4: Cross-domain thresholding (b=books, d=dvd, e=electronics, k=kitchen), no voting.

Factoring out the hard cases using accuracy-based thresholding leads to significant accuracy gains for all four domains. Also, comparing these results to the results of Blitzer et al. (2007), we notice accuracy gains. F-score based thresholding yields low values of OOD, and accuracies that exceed the best results of Blitzer et al. (2007): for books, we obtain an accuracy of 83 (against 80.4 by Blitzer et al. (2007)), for dvd, we obtain 82.8 (against 82.4), for electronics 86.8 (against 84.4) and for kitchen 89 (against 87.7). For cross-domain thresholding, we observe significant accuracy gains compared to uncalibrated cross-domain classification for accuracy-based thresholding, at the expense of relatively low yield. Yield increases however significantly for F-score based thresholding, while preserving the accuracy gains.

Comparing our cross-domain calibration results to the results of Blitzer et al. (2007), we observe that our results when optimizing for accuracy are better throughout,

θ_{Acc}	% OOD	Acc	P
books	45.5	82.8	65.8
dvd	41.8	85.8	69.4
electronics	37.3	81	70.7
kitchen	47.5	90	66.3
θ_F	% OOD	Accuracy	P
books	23.3	80.8	78.7
dvd	6	79.8	86.3
electronics	7	77.3	84.4
kitchen	8	85.5	88.6
θ_{yield}	% OOD	Accuracy	P
books	0.3	72.5	84
dvd	0.5	77.5	87.1
electronics	0.5	73.8	84.7
kitchen	0.5	83.3	90.7

Table 5: Cross-domain thresholding, voting.

of course with the drawback of a large portion of out of domain data: we obtain an average increase of 3.75% in accuracy. For F-score based thresholding, we observe improvement in 3 out of 8 cases: for kitchen to books, we obtain 81.5 against 73.2 of Blitzer et al. (2007); for books to dvd, we have 79.8 against 78.5, and for kitchen to dvd, we have 79.3 against 76. Table 5 lists results obtained by majority voting over classes. Compared to the non-voting results in Table 4, classifier combination appears effective for all three types of thresholding.

Conclusions

We have outlined a novel procedure for multi-domain sentiment classification based on cross-domain classifier calibration. Using a small fragment of annotated data from a new domain, we estimated thresholds for classifiers based on existing training data and this new data fragment. We proposed three performance measures for finding these thresholds: accuracy, F-score and yield, and demonstrated that these measures allow for workflow-oriented trade-offs between classifier performance and human effort. F-score based thresholding appears a good compromise between accuracy and manual inspection: accuracies obtained are relatively high, outperforming in-domain classification, and the amount of data that can be analyzed automatically is relatively large. From the perspective of a practical workflow, adding a new sentiment domain to an existing multi-domain classifier (which, in its simplest form, consists of a set of single-domain classifiers) would involve annotating a small portion of new data, computing thresholds for all existing sentiment domains with respect to this new data, and combining results through majority voting. In future work, we intend to investigate the use of transductive learners to alleviate the need for manual annotation of out of domain data.

References

Andreevskaia, A., and Bergler, S. 2008. When specialists and generalists work together: Overcoming domain dependence in sentiment tagging. In *Proceedings of ACL-08: HLT*, 290–298. Columbus, Ohio: Association for Computational Linguistics.

Aue, A., and Gamon, M. 2005. Customizing sentiment classifiers to new domains: A case study. In *Proceedings of Recent Advances in Natural Language Processing (RANLP)*.

Ben-David, S.; Blitzer, J.; Crammer, K.; Kulesza, A.; Pereira, F.; and Vaughan, J. 2009. A theory of learning from different domains. *Machine Learning Journal*.

Blitzer, J.; Dredze, M.; and Pereira, F. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of ACL-07*.

Chang, C.-C., and Lin, C.-J. 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Kraaij, W.; Raaijmakers, S.; and Elzinga, P. 2008. Maximizing classifier utility for a given accuracy. In *Proceedings of BNAIC-2008*, 121–128.

Li, S., and Zong, C. 2008. Multi-domain sentiment classification. In *HLT '08: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, 257–260. Morristown, NJ, USA: Association for Computational Linguistics.

Li, T.; Sindhwani, V.; Ding, C.; and Zhang, Y. 2009. Knowledge transformation for cross-domain sentiment classification. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 716–717. New York, NY, USA: ACM.

Raaijmakers, S., and Kraaij, W. 2009. Polarity classification of Blog TREC 2008 data with geodesic kernels. In *Proceedings TREC 2008, Gaithersburg, USA*.

Raaijmakers, S. 2009. *Multinomial language learning, Investigations into the geometry of language*. Ph.D. Dissertation, Tilburg University.

Teufel, S. 2007. An overview of evaluation methods in TREC ad hoc information retrieval and TREC question answering. In *Evaluation of Text and Speech Systems*, 163–186.

Wu, Q.; Tan, S.; and Cheng, X. 2009. Graph ranking for sentiment transfer. In *ACL-IJCNLP '09: Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, 317–320. Morristown, NJ, USA: Association for Computational Linguistics.

Zhang, D.; Chen, X.; and Lee, W. S. 2005. Text classification with kernels on the multinomial manifold. In *Proceedings of SIGIR '05*, 266–273.