Efficient Cross-Domain Classification of Weblogs

Elisabeth Lex
Know-Center
elex@know-center.at

Michael Granitzer
Know-Center
mgrani@know-center.at

Andreas Juffinger
The European Library
andreas.juffinger@kb.nl

Christin Seifert
Know-Center
cseifert@know-center.at

Abstract

Text classification is one of the core applications in data mining due to the huge amount of uncategorized digital data available. Training a text classifier results in a classification model that reflects the characteristics of the domain it was learned on. However, if no training data is available, labeled data from a related but different domain might be exploited to perform cross-domain classification. In our work, we aim to accurately classify unlabeled weblogs into commonly agreed upon newspaper categories using labeled data from the news domain. The labeled news and the unlabeled blog corpus are highly dynamic and hourly growing with a topic drift, so the classification needs to be efficient. Our approach is to apply a fast novel centroid-based text classification algorithm, the Class-Feature-Centroid Classifier (CFC), to perform efficient cross-domain classification. Experiments showed that this algorithm achieves a comparable accuracy than k-NN and Support Vector Machines (SVM), yet at linear time cost for training and classification. We investigate the classifier performance and generalization ability using a special visualization of classifiers. The benefit of our approach is that the linear time complexity enables us to efficiently generate an accurate classifier, reflecting the topic drift, several times per day on a huge dataset.

1. Introduction

The automatic classification of natural language texts has become one of the most important applications in data mining, especially due to the steadily growing amount of unstructured digital content available on the Web. In general, text classifiers are constructed by learning characteristic patterns from already labeled documents[1]. However, learning on a specific domain generates a classifier model that reflects the patterns of exactly the domain the classifier was trained on. In context of text classification, especially the vocabulary is important and significant to distinguish between classes because in the vector space model (VSM), each unique term corresponds to a feature in the vector space. Naturally, each classifier trained on this vector space heavily depends on the terms and the vocabulary used in this domain. Also, classifiers usually need a lot of training data in order to fully capture a domain's characteristics.

When no training data is available for a domain, a possible approach is to train a classifier on a domain with similar features and characteristics. However, it is not clear whether a trained model generated on one domain can be generalized to another domain. Classifiers usually try to avoid over-fitting on the training set to maximize the generalization capabilities. Clearly, in case of cross-domain classification, an implicit fitting to the training corpus vocabulary is unavoidable. One approach to overcome this is to exploit the latent semantic structures of texts. Based on this, a more domain independent latent semantic feature space can be created. Also, the semantic relations between different domains can be exploited to transfer knowledge from a source domain to a target domain. In [2], an algorithm based on Latent Semantic Analysis (LSA) [3] is proposed to exploit common topics and their semantic relations between domains. This knowledge is then used to boost text classification in the target domain. The algorithm performs well, yet its computational complexity is very high. In [4], an algorithm is proposed that explicates the latent semantic relations between two domains in combination with a co-clustering approach. However, this method is also computationally complex and therefore hardly applicable for a highly dynamic setting.

In our work, we aim to classify weblogs (further referred to as blogs) into commonly agreed upon newspaper categories. In general, there is usually little training data available for blogs, even though blogs are often tagged by users. Yet, this tagging information is mostly subjective and not consistent across blogs [5], and most important, the tagging information does not match the predefined newspaper

categories. Besides, the tagging vocabulary of each individual blog is different and dynamically changing. These are the reasons why we cannot exploit folksonomies[1] either. We would need to first learn a mapping from the evolving tagging vocabulary or folksonomy to our categories which would result in another text classification problem. Consequently, it is not possible to directly exploit individual blog tagging information, folksonomies respectively in our text classification setting.

Besides, in our project, we already are provided with high quality labeled articles from a German news corpus [6]. The articles were manually assigned to well known and commonly agreed upon newspaper categories by newspaper editors.

Also, the blogs contained in our blog corpus are quite similar to news articles [7], because in our project setting, we are primarily interested in blogs highly correlated with news articles. An analysis of the term distributions of both corpora revealed significant differences, yet, we also identified that significant terms for a topic remain the same across both corpora. This is in fact a crucial property for cross-domain text classification. Given the labeled news articles and the unlabeled blog corpus, the question is: can we use the news data to apply high quality cross-domain classification from news to blogs? Our corpora are highly dynamic and daily growing with a topic drift. Therefore, a fast and efficient solution is needed for our work.

To derive an efficient solution, we apply several text classification algorithms, among others a novel centroid-based algorithm, on the problem setting and evaluate the performance of these algorithms for different scenarios. Note that our methodology was first introduced in [8].

We claim that the generalization abilities of text classification algorithms are sufficient when the classifiers implicitly concentrate only on the most important textual features. Our features are weighted with state-of-the-art techniques. In addition to the performance evaluation, we conduct a visual evaluation with the classifier visualization proposed in [9]. This visualization enables us to investigate classifier decisions and mis-classifications in more detail and to deeply analyze the generalization abilities of the algorithms on the different scenarios.

The remainder of this paper is structured as follows: Section 2 describes the cross-domain problem setting of our work and the centroid-based classifier we used among others. In Section 3, the classifier visualization is described and its application to our classification task is explained. Section 4 describes the experimental settings, our data sets and the parameter settings. In Section 5, our results are described and discussed. Finally, we conclude the work in Section 6 and give directions for future work.

[1] *http://www.iskoi.org/doc/folksonomies.htm*

2. Problem Setting

In our problem setting, the goal is to solve a cross-domain multi-class problem with five classes. We are provided with a corpus that consists of two sub corpora: The first sub corpus is collected from Austrian newspapers. We refer to this corpus as news corpus. Note that newspaper editors manually labeled the corpus according to five common newspaper categories: politics, economy, sports, culture and science. The second sub corpus was crawled from the World Wide Web using a Web crawler developed at our institution [10]. We further refer to this sub corpus as blog corpus.

For our project setting, it is crucial to be provided with a classification system with low computational complexity due to the highly dynamic nature of our data. Consequently, it is natural to use a centroid-based text classifier because centroid-based classifiers are known to achieve good results in terms of accuracy and time complexity [11, 12]. In this work we implemented a novel centroid-based text classifier, the Class Feature Centroid (CFC), recently introduced in [13]. The algorithm is described in more detail in Section 2.1. The CFC is extremely fast and is reported to outperform SVM – which is one of the best performing text classifiers - and all other centroid based text classifiers.

To compare the performance of the cross-domain classification task, we evaluated CFC, and two standard text classifiers: Firstly, a k-Nearest Neighbor algorithm (k-NN) [14] and secondly, a Support Vector Machine (SVM) [15] based on LibLinear [16]. As outlined by Sebastiani [1], SVMs and k-NN algorithms are among the best performing standard text classification algorithms.

2.1. CFC Algorithm

The Class Feature Centroid (CFC) classifier, proposed by Guan et al. in [12], is a novel approach to centroid-based text classification. The advantage of centroid-based classifiers is clearly their computational efficiency and their simplicity [11]. However, their accuracy strongly depends on the quality of the centroids representing each class [12]. In addition to the CFC algorithm, in [13] a novel centroid weight representation is proposed which takes into account both the inter-class term distribution and the inner-class term distribution. Both are then combined to generate the term weights for the centroids. The weight representation is given in Formula (1)

$$w_{ij} = b^{\frac{DF_{t_i}{}^{j}}{|C_j|}} \times \log\left(\frac{|C|}{CF_{t_i}}\right) \quad (1)$$

For each class, a centroid vector is constructed using this weighting scheme. Since only highly discriminative terms result in a weight larger than 0, highly discriminative centroids are derived. In addition to this weighting, a denormalized cosine

similarity is used to compute the distance of the test documents to the class centroids. The denormalized cosine similarity computes the similarity between a document vector and a centroid vector using a standard cosine similarity whereas the centroids are not normalized to unit length as in tf-idf. The authors claim that not normalizing the centroids to unit length preserves the discriminant capabilities of the centroids.

Apart from the novel weighting representation, the memory consumption of the CFC is very low because one has only to store one highly sparse centroid vector per class. Guan et al. also compared the algorithm's performance to a SVM and other centroid-based approaches whereas the experiments were conducted on the Reuters-21578 corpus and on the 20-newsgroup corpus. In these experiments, the algorithm outperforms both the SVM and other centroid-based approaches.

3. Classifier Visualization

To get more detailed insights into a classifier's quality and decisions, we used the visualization of classifiers proposed in [9]. The visualization is applicable for all classifiers whose output corresponds to or can be mapped to an a-posteriori distribution. Our k-NN implementation outputs probabilities per se, for SVM and CFC, the classification outputs need to be mapped to a probability distribution, as described by Platt in [17]. The visualization shows the quality and decisions of a classifier for a fixed split of the data set. In practical applications, this split is given beforehand, since only a subset of the data contains samples with class information (so-called labeled samples).

In our scenario, all items exhibit class information. Thus, we randomly split the data set into training, test and evaluation subsets. Figure 1 shows the general visualization process using these three subsets.

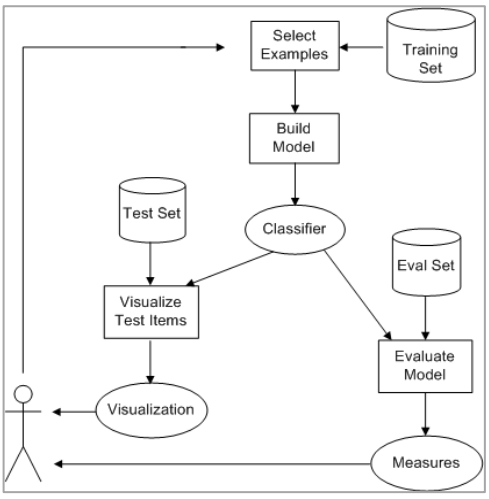

Figure 1. General visualization process.

In the visualization, the classes are represented by unique colored squares that are placed equally distributed around the perimeter of a circle. As shown in Figure 1, the classifier is learned on the training items and visualized on the test set whereas the visualization shows the a-posteriori probabilities for all test items. Items with low confidence are placed in the middle of the circle and items with high confidence are placed near their according class squares. Consequently, the classifier visualization provides us with the possibility to interactively identify problematic classes and items at a glance.

Additionally, common performance measures like accuracy or precision/recall are calculated and provided in form of a tabular view and a performance plot. With the performance measures, we are able to compare different algorithms and rank them in terms of, for instance, accuracy. Besides, if the correct class label information is available also for the test set, the mis-classifications of the classifier on the test set can be investigated in detail. In the mis-classification view, correctly classified items are represented by a green "+"and mis-classified items are denoted by a red "x". Naturally, the mis-classification view can be used to indicate problem samples and to identify classes which are hard to separate.

In general, when evaluating classification results, especially the generalization abilities of classifiers are of great interest. If a classifier exhibits no generalization abilities at all, it cannot be applied to another domain. This is referred to as overfitting, which means that the classifier adapts the classification model to exactly the used training data. Naturally, if the test set differs from the training set, the application of such a classifier results in a bad performance. The overfitting behavior of a classifier can be assessed by comparing the confusion matrix of the training set with the confusion matrix of the test set. A confusion matrix shows for each pair of classes how many items of the one class were assigned to the other class. The diagonal of the confusion matrix contains the correctly classified items. When analyzing the confusion matrix of the classifier performance on the evaluation set, a classifier with good generalization abilities should output a similar confusion matrix like on the training set. Note that in order to compare both matrices, it is necessary to normalize them to the number of documents in the respective data set.

In our work, we visually analyze the overfitting behavior of a classifier using a heatmap-based visualization of the confusion matrix. Note that different basis colors are chosen for the training and the evaluation set. The cell containing the maximum value is colored using the darkest color whereas the cell with the minimum value is colored white. Consequently, two confusion heatmaps can be visually compared because the absolute value in the cell is irrelevant. This is important as in general, the number of items in the training and the test set differ to a great extent.

4. Experimental Settings

Four our experiments and the evaluation of our classification task, we split our data sets into a fixed training and test set. To measure the performance of our classifiers, we then evaluated the algorithms on the following four scenarios:

NB NewsNews: The training set of the news corpus was used to train the classifiers and we report the performance on the news evaluation set.

BB BlogBlog: The training set of the blog corpus was used to train the classifiers and we report the performance on the news evaluation set.

NB NewsBlog: The training set of the news corpus was used to train the classifiers and we report the performance on the blog evaluation set.

BN BlogNews: The training set of the blog corpus was used to train the classifiers and we report the performance on the news evaluation set.

4.1 Dataset Properties

The news corpus contains about 28k documents with about 237k nouns in total and an average document length of 92.5 nouns. Each class of the news corpus contains nearly the same number of documents (~5600).
The blog corpus consists of about 11k blog entries from 56 blogs which are selected according to the given newspaper sections: 10 politics blogs, 10 economy blogs, 10 sports blogs, 11 culture blogs, and 15 science blogs. In the blog corpus, the classes are represented by about 2800 politic blog entries, 2800 economy blog entries, 2400 sports blog entries, 1400 science blog entries, and 1100 culture blog entries. The blog entries were labeled with the class of the blog, and therefore all blog entries within a single blog belong to the same class. The blogs were chosen by their news categories and their blog entries were randomly checked whether they really belong to the class. Note that we did not examine all blog entries per blog therefore we cannot completely ensure that there is no mislabeled data. This naturally limits the theoretically achievable accuracy to less than 100%. The blog corpus contains about 110k nouns with an average document length of 61.5 nouns and a total noun token count of 675k. The merged corpora dictionary holds 302k different terms. Note that the sum of the distinct terms in the news and blogs dictionary is 347k, so consequently these two corpora share only 45k terms. We vectorized the data with an information extraction and vectorization module based on OpenNLP[2]. We used the Part-Of-Speech tags to construct our noun vector space. As a measure for the statistical difference between the news and blog dataset, we calculated the Kullback-Leibler divergence (KL) [18]. The Kullback-Leibler divergence, also known as relative entropy, is a measure of the difference between two probability distributions B and N. The KL divergence between two corpora (Blog B, News N) is calculated as

$$KL\ (B\ \|N) = \sum_t \left[P_B(t) \log\left(\frac{P_B\ (t)}{P_N\ (t)}\right)\right] \quad (2)$$

where $P_{B(t)}$ states the probability of the term t in corpus B and $P_{N(t)}$ the probability in corpus N. The Kullback Leibler divergence for our cross domain corpora is shown in Table 1. The KL divergence of 0.535 clearly reflects the statistical difference between the term statistics of these two corpora.

Table 1. Kullback Leibler divergence for blog versus news corpus.

	BN	NB	Mean
KL Global	0.535	0.430	0.483

Note that the number of documents per class is not necessarily equally distributed across the different sets (train/test) due to the purely random split procedure.

4.2 Parameter Settings

For a weighting schema, we used BM25 [19] for k-NN and SVM with the standard parameters k=2 and b=0.75. We also experimented with variants of tf-idf for both algorithms, yet the k-NN and SVM algorithms performed best with BM25.
For CFC, we used a standard tf-idf weighting, as recommended by the authors. We also tested the algorithm with BM25, but the results got worse than with tf-idf, as expected. For the k-NN algorithm, we conducted a manual parameter search and identified k=10 to be the best parameter setting. For the SVM, we used a linear kernel which is reported to outperform non-linear kernels in text classification [20]. We also experimented with various values for parameter b in CFC. However, different from findings in the original publication, where $b=e^{-1.7}$, we found that $b=e^{-1.0}$ performs best in our problem setting.

5. Results and Discussion

In our evaluations, we first computed the performance of all three classifiers on the mono-domain classification task (scenarios NN and BB). From these experiments, we determined the

[2] http://opennlp.sourceforge.net

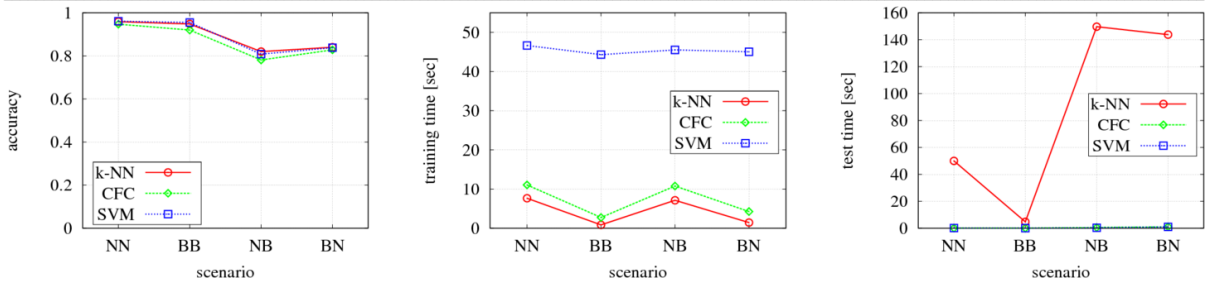

Figure 2. Accuracy and computation time for all algorithms and scenarios.

maximum achievable performance for all classifiers for the cross-domain task.

As a performance measure, we computed the micro-averaged classifier accuracy a using the following formula:

$$a = \frac{\sum TP + \sum TN}{\sum TP + \sum FP + \sum FN + \sum TN} \quad (3)$$

Note that all our results are derived following a 10-fold crossvalidation strategy.

Our experiments revealed that the CFC achieves an accuracy value of 0.95 in the mono-domain task, equally well as the accuracy of the k-NN. The SVM performs slightly better with an accuracy of 0.96. We achieved similar results for scenario BB. The experiments showed that in the mono-domain settings, all algorithms achieve a nearly perfect performance. Since our computations are based on nouns which capture mostly topic related information, we can conclude that generally, a topic classification can be done with high accuracy in one domain.

For scenario NB, the most important scenario for our work, the performance of all three classifiers drops significantly, as expected. The k-NN algorithm is best with accuracy of 0.82. The CFC algorithm performs with accuracy of 0.78, and the SVM with accuracy of 0.80. The standard deviations for the 10 fold crossvalidation range from 0.001 to 0.009. Since this is extremely low this indicates that the classifier performance is robust and independent from the actual splits. Additionally, we reduced this scenario NB to a binary classification task, taking only news articles and blog entries from the classes “politics” and “sports” into account. The results for the binary task versus the five class problem are: for the k-NN 0.85 versus 0.80, for the SVM: 0.82 versus 0.76, and for the CFC: 0.84 versus 0.78. Note, in this scenario, the accuracy dropped less from scenario NN to scenario NB, clearly due to the lower complexity of the binary task.

For completeness of our experiments, we also evaluated scenario BN. The k-NN algorithm performs slightly better (accuracy of 0.84) than CFC with accuracy of 0.838. The cross-domain experiments reveal that a topic classification based on nouns is harder across two different domains, as expected. Clearly, the vocabulary of two domains is different and so are obviously the topics, at least to a certain extent.

When comparing the results of the different classifiers, we can derive that in many cases, the k-NN algorithm works slightly better than the CFC and the SVM. All results are summarized in Table 2 whereas the accuracy means and standard deviation of the 10-fold crossvalidation are given for all algorithms and both the mono-domain and the cross-domain scenarios.

Table 2. Accuracy for all scenarios and algorithms (mean and standard deviation).

	NN	NB	BB	BN
k-NN	0.959 ± 0.005	**0.820 ± 0.004**	0.948 ± 0.001	**0.840 ± 0.005**
LibLinear	**0.961 ± 0.005**	0.808 ± 0.006	**0.955 ± 0.007**	0.838 ± 0.004
CFC	0.947 ± 0.009	0.781 ± 0.004	0.920 ± 0.009	0.828 ± 0.005

Regarding the computation time, the CFC is by far the best. In Figure 2, the training and testing time for all investigated algorithms and scenarios are given. As one can see, the CFC is slightly slower than the k-NN in training, but faster in testing. The training time of the SVM is worse even though its testing time is similarly low as the testing time of the CFC. Summing up, taking both training and testing time into account, the CFC performs best out of all three algorithms.

To get deeper insights into the algorithm's decisions, we analyzed the class centroids of CFC in more detail. For this, we investigated the terms with weight w>0 (remember, these terms are claimed to be the most discriminant terms, as described in Section 3). We calculated the KL divergence between the news and blog term distributions, but for this experiment, we only took centroid terms into account. As shown in Table 3, the KL divergence significantly decreased (by a factor 4 on average). This reveals that the CFC selects those terms which are characteristic for a class and these remain the same across both corpora. This indicates that the centroid weights are robustly selected by the CFC. Note that C1 corresponds to class politics, C2, to class economy, C3 to sports, C4 to culture, and C5 to science.

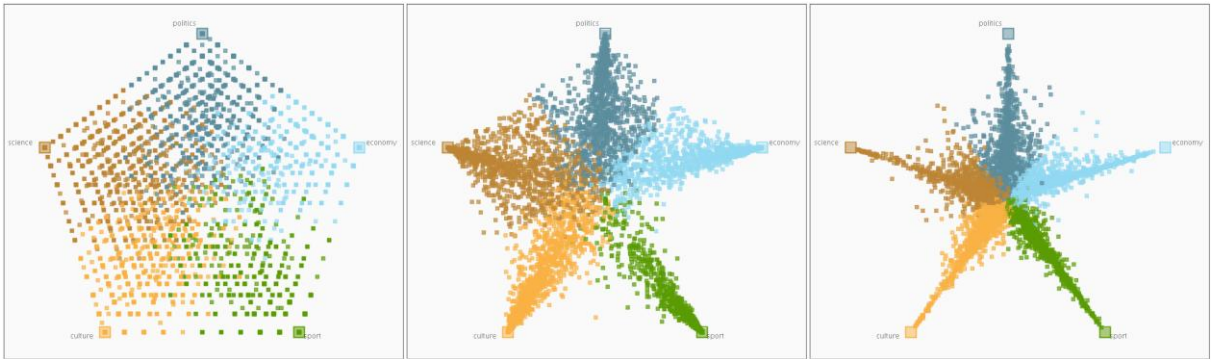

Figure 3. Visual analysis of scenario NN: k-NN, SVM, and CFC.

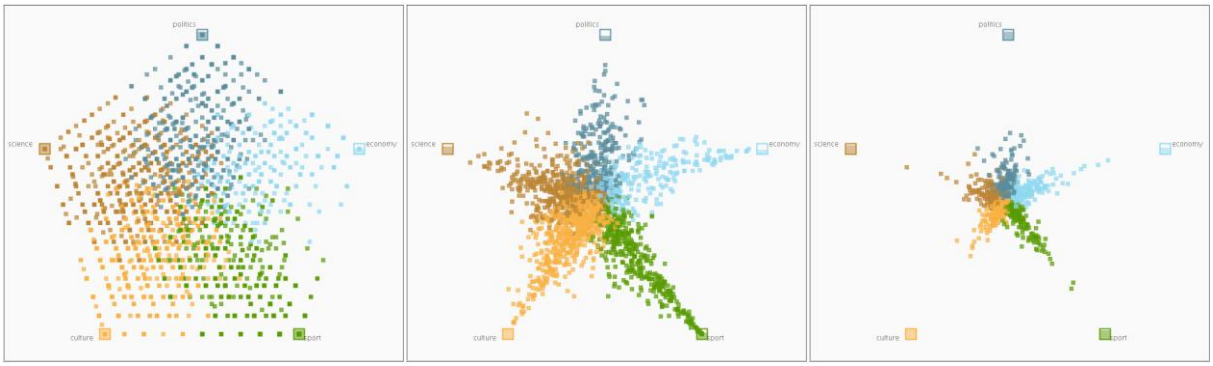

Figure 4. Visual analysis of scenario NB: k-NN, SVM and CFC.

Table 3. KL divergence for CFC centroids

	BlogNews	NewsBlog	Mean
KL Local C_1	0.048	0.065	0.056
KL Local C_2	0.127	0.194	0.160
KL Local C_3	0.216	0.117	0.166
KL Local C_4	0.081	0.067	0.074
KL Local C_5	0.123	0.127	0.125
KL Local $C_{Ø}$	0.119	0.114	0.116

We visually analyzed the classification results for scenario NB with our visualization tool. The results are depicted in Figure 3 and 4. The visualization shows that the k-NN algorithm exhibits a discrete probability distribution. This can be derived from the fact that between two classes a maximum of 11 discrete confidence steps can occur. This also holds for the multi-class assignment. However, an interesting artifact of our dataset is that the k-NN has difficulties to make a clear decision between the classes “science” and “culture” as well as “economy” and “sports”. Also, the visualization gives the impression that the SVM analyzes a set of binary classification problems since the test items are placed along the connecting lines between all classes. However, we need to investigate this artifact in more detail. The CFC algorithm exhibits a relatively balanced distribution of the test items. Comparing the image of SVM and CFC, the CFC places less test items on the outer boundaries, between the categories “politics” and “economy”, as well as the categories “politics” and “science”. The reason for this is that the centroid vectors overlap to a certain extent.

The visualization also reveals that the CFC does not prefer any class. This better reflects the a-priori probabilities because we train on an equally distributed corpus. For the NB scenario, k-NN and CFC exhibit a very similar visual distribution – like in the NN scenario. That is why we expect that the algorithms perform similar on the cross-domain task. For the CFF and the SVM one can clearly see the worse performance for classes in the cross-domain task. For the CFF and the SVM one can clearly see the worse performance for classes in the cross-domain task. The accuracy of the classifier is still quite high yet the confidence in the decision is still decreased (most items are located in the center of the circle which means their confidence is low). However, the correctness of the algorithms' decisions can only be verified when investigating the mis-classifications (as depicted in Figure 4). The mis-classification of k-NN is equally distributed as one can see in Figure 3 (a). The SVM has several mis-classifications in category “science” with confidence nearly 1. This indicates that some of the support vectors cannot be generalized from the news domain to the blog domain.

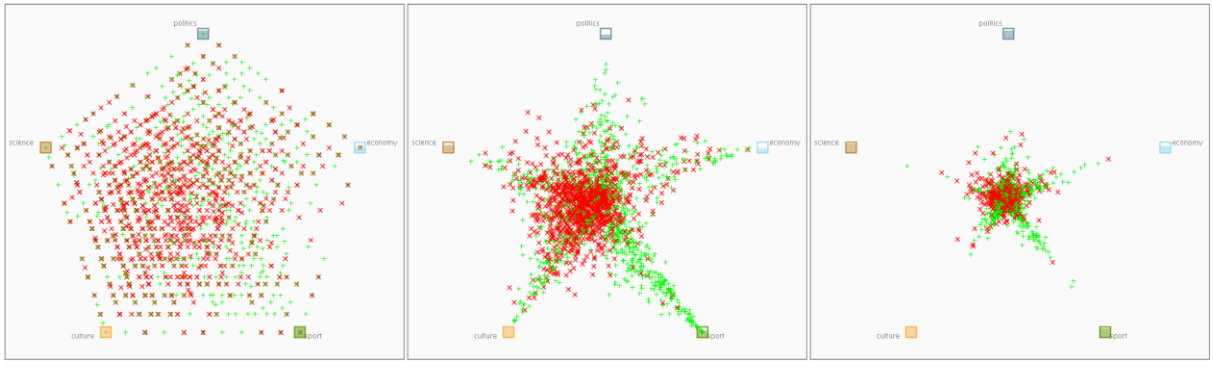

Figure 5. Mis-classification view of scenario NB.

The visual impression is reflected in the accuracy result. In contrary, the CFC has nearly no mis-classification very close to the classes whereas the actually misclassified items are placed more in the center of the visualization. Remember that this is the region where the items with low classifier confidence are placed, as described in Section 3. In reference to our dataset, we conclude that the CFC confidences are more trustworthy than the SVM and k-NN confidence values for the cross-domain task, since the CFC places misclassified items rather in the middle of the circle.

Additionally, we visually investigated the generalization abilities of our classifiers in the cross-domain setting. For this experiment, we also used the classifier visualization to generate heatmaps that encode the confusion matrix of the classifiers. We refer to them as confidence heatmaps. The confidence heatmaps are shown in Figure 6 for scenario NB and NN. In the mono-domain task (NN, left in Figure 5) the confidence maps for the training and evaluation set are quite similar. This indicates that all classifier exhibit a good generalization behavior. In the cross-domain task (NB, right of Figure 5), the evaluation maps show darker colored off-diagonal elements whereas the training heatmap reveals a distinct diagonal. From this, it can be derived that the classifiers overfit towards the training data in the cross-domain task which means that their generalization abilities are lower than in the mono-domain task.

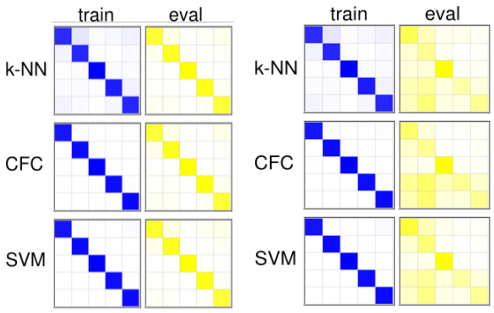

Figure 6. Confidence heatmaps (left NN, right NB).

7. Conclusions

In this publication, we applied and evaluated a novel linear time classifier (CFC) for a cross-domain classification task. Our experiments showed that this classifier performs comparably with SVM and k-NN while being remarkably faster. Also, in terms of memory complexity the CFC definitely outperforms SVM and k-NN. In our setting, the CFC only stores the five centroid vectors, the SVM has to store about 1000 support vectors, and the k-NN the full training set (17k for scenario NN and NB). We also identified that in our setting, the CFC model is more general and therefore better applicable than SVM for the cross-domain task. Besides, the experiments revealed that the accuracy drops less for CFC from the mono-domain task to the cross-domain task. This also emphasizes that the CFC model generalizes better. The visualization shows that the confidence distribution for CFC is more trustworthy than the SVM and k-NN confidence values. The k-NN mis-classifies items to wrong classes with high confidences, significantly more often than CFC and SVM. Furthermore, the CFC and SVM rather assign low confidence values to misclassified items. This clearly reflects the decision uncertainty for such items. However, when analyzing the computation time and taking into account both training and testing time, the CFC outperforms both the k-NN and the SVM. To sum up, in our case, the CFC is an efficient solution for a cross-domain classification of weblogs. A visual analysis of the overfitting behavior of the used classifiers with confidence heatmaps revealed that in the cross-domain task, all classifiers tend to adapt their model to exactly the training data. This naturally limits their generalization abilities. In the future, we want to analyze whether incremental algorithms are applicable for our problem setting and how to improve the generalization abilities of our classifiers. Also, we plan to extend our blog set to English sources in order to address a more international setting. Besides, we want to investigate which textual and style based features are especially suited for a cross-domain classification that goes beyond topic classification.

10. References

[1] F. Sebastiani, "Machine learning in automated text categorization", *ACM Comput. Surv.*, vol. 34, no. 1, pp1-47, 2002.

[2] G., Xue, W. Dai, Q. Yang, Y. and Yu, "Topic-bridged PLSA for cross-domain text classification", in *Proc. Int. ACM Conf. on Research and Development in Information Retrieval (SIGIR)*, New York, NY, USA, 2008, pp. 627-634.

[3] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, „Indexing by latent semantic analysis", *Journal of the American Society for Inf. Science*, vol. 41, 1990.

[4] P. Wang, C. Domeniconi, and J. Hu, "Using Wikipedia for co-clustering based cross-domain text classification", in *Proc. IEEE Int. Conference on Data Mining (ICDIM)*, Washington, DC, USA, 2008.

[5] U. Farooq, T. Kannampallil, Y. Song, C. Ganoe, J. Carroll, and L. Giles, "Evaluating tagging behavior in social bookmarking systems: metrics and design heuristics", *Proc. Int. ACM Conf. on Supporting Group Work (GROUP)*, New York, NY, USA, 2007, pp. 351-360.

[6] E. Lex, C. Seifert, W. Kienreich, and M. Granitzer, „A generic framework for visualizing the news article domain and its application to real-world data", *Journal of Digital Information Management*, 2008, vol. 6, no. 6, pp. 434-442.

[7] A. Juffinger, M. Granitzer, and E. Lex, "Blog credibility ranking by exploiting verified content", in *Proc. Workshop on Information Credibility on the Web (WICOW)*, New York, NY, USA, 2009, pp. 51-58.

[8] E. Lex, C., Seifert, M., Granitzer, and A., Juffinger, "Cross-Domain Classification: Trade-Off between Complexity and Accuracy", *Proceedings of the 4th International Conference for Internet Technology and Secured Transactions (ICITST)*, 2009.

[9] C. Seifert, and E. Lex, "A Novel Visualization Approach for Data-Mining-Related Classification, in *Proc. Int. Conference on Information Visualization (IV)*, 2009.

[10] A. Juffinger, T. H. Neidhart, M. Granitzer, R. Kern, A. Weichselbraun, G. Wohlgenannt, and A. Scharl, „Distributed web2.0 crawling for ontology evolution", in: *International Journal of Internet Technology and Secured Transactions*, 2008.

[11] E.-H. Han, and G. Karypis, "Centroid-based document classification: Analysis and experimental results", in *Proc. European Conf. on Principles of Data Mining and Knowledge Discovery (PKDD)*, London, UK, 2000, pp. 424-31.

[12] V. Lertnattee, and T. Theeramunkong, "Effect of term distributions on centroid-based text categorization", *Information Sciences, Informatics and Computer Science*, vol. 158, no. 1, 2004.

[13] H. Guan, J., Zhou, and M. Guo, "A class-feature-centroid classifier for text categorization", in *Proc. Int. Conf. on World Wide Web (WWW)*, New York, NY, USA, 2009.

[14] D. Aha, D. Kibler, and M. Alber, "Instance-based learning algorithms", *Machine Learning*, vol. 6, no. 1, 1991, pp.37-66.

[15] V. Vapnik, "The Nature of Statistical Learning Theory", *Springer*, 1995.

[16] R. E. Fan et al, "LIBLINEAR: A Library for Large Linear Classification", Journal of Machine Learning Research, vol. 9, pp. 1871 – 1874, 2008.

[17] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods", in *Advances in Large Margin Classifiers*, 1999.

[18] S. Kullback, and R. Leibler, "On information and sufficiency", *Annals of Mathematical Statistics*, vol. 22, pp. 79-86, 1951.

[19] K. Sparck Jones, S. Walker, and S. E. Robertson, "A probabilistic model of information retrieval: development and comparative experiments", *Inf. Process. Management, Research and Development in Information Retrieval (SIGIR)*, New York, NY, USA, pp. 42-49, 1999.

[20] Y. Yang, and X. Liu, "A re-examination of text categorization methods", in *Proc. Int. ACM Conf. On Research and Development in Information Retrieval (SIGIR)*, New York, NY, USA, pp. 42-49, 1999.

11. Acknowledgements

The Know-Center is funded within the Austrian COMET Program – Competence Centers for Excellent Technologies - under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.