Content-Based Trust and Bias Classification via Biclustering*

Dávid Siklósi Bálint Daróczy András A. Benczúr
Institute for Computer Science and Control, Hungarian Academy of Sciences
{sdavid, daroczyb, benczur}@ilab.sztaki.hu

ABSTRACT

In this paper we improve trust, bias and factuality classification over Web data on the domain level. Unlike the majority of literature in this area that aims at extracting opinion and handling short text on the micro level, we aim to aid a researcher or an archivist in obtaining a large collection that, on the high level, originates from unbiased and trustworthy sources. Our method generates features as Jensen-Shannon distances from centers in a host-term biclustering. On top of the distance features, we apply kernel methods and also combine with baseline text classifiers. We test our method on the ECML/PKDD Discovery Challenge data set DC2010. Our method improves over the best achieved text classification NDCG results by over 3–10% for neutrality, bias and trustworthiness. The fact that the ECML/PKDD Discovery Challenge 2010 participants reached an AUC only slightly above 0.5 indicates the hardness of the task.

Categories and Subject Descriptors

H.3 [**Information Systems**]: Information Storage and Retrieval; I.2 [**Computing Methodologies**]: Artificial Intelligence; I.7.5 [**Computing Methodologies**]: Document Capture—*Document analysis*

General Terms

Biclustering, Co-clustering, Feature Selection, Document Classification, Information Retrieval

Keywords

Web Quality, Trust, Bias, Machine Learning, Document Classification

1. INTRODUCTION

Mining opinion from the Web and assessing its quality and trustworthiness became a well-studied area [12]. Known results typically mine Web data on the micro level, analyzing individual pages of blogs or even sections containing comments and reviews.

Our aim is to address a slightly different tasks of assessing trust and neutrality on the high level. Our purpose is to aid the first step of information gathering: to select relevant and trustworthy subcorpora, and to aid archival institutions to maintain such large scale collections over extended periods of time. Hence we cannot rely on the heavy machinery of opinion mining and sentiment analysis [24], methods that will likely not scale to the Web size.

Host level classification is typically based on simple features such as tf.idf for relative small vocabularies or content and linkage based information first compiled for spam classification [5]. The same task for classifying various aspects of quality on the host level were, to our best knowledge, first introduced as part of the ECML/PKDD Discovery Challenge 2010 tasks. Participants [18, 2, 23] found the new tasks of neutrality, bias and trust particularly challenging with AUC values, in all cases, below 0.6, typically even near the 0.5 value of a completely random prediction.

Based on our findings on the ECML/PKDD Discovery Challenge 2010 data set [14] where a random forest classifier based on the top few 10,000 terms performed best for neutrality, bias and trust, we will give improved text classification techniques specifically designed or these tasks. The key ingredients of our method are as follows.

- We compile around 1000 bags of concepts from words via biclustering. This low dimensional representation allows computationally costly classifiers, in particular SVM, to be applied. Important to note that unlike in the original biclustering method [13] that uses Kullback-Leibler, we use Jensen-Shannon divergence that greatly improves the quality of the final prediction.
- We use simple feature selection and weighting based on frequencies in the training set. Unlike for all other categories of spam and genre, this method is particularly suited to the highly imbalanced classes of non-neutrality, bias and distrust.
- Given the compact representation of hosts by cluster distances, we may apply computationally expensive methods for classification. We use SVM with several kernels and apply late fusion.

The idea of representing objects by cluster distance vectors originates from image classification [25] where Gaussian mixtures are used as a soft clustering method over image descriptors that are aggregated into feature vectors of the images. The combination of various SVM kernels apply very well for

*This work was supported by the EU FP7 Project LAWA: Longitudinal Analytics of Web Archive Data.

WebQuality '12, April 16, 2012, Lyon, France

classification over these features [11] and for other tasks as well [26].

We compare our result over the ECML/PKDD Discovery Challenge 2010 data set, both with the best results of the participants [18, 2, 23] and with our earlier results focusing primarily on spam classification [14]. Our improvements in NDCG for bias is 20% over DC2010 best and 5% over [14]. For distrust we gain 20% and 10%, respectively, while for non-neutrality 3%. The AUC values of our various methods are convincingly above 0.6, i.e. we may say that we have reached the quality of the first results on Web spam a few years ago.

The rest of this paper is organized as follows. First we begin with an extended motivation of our new text classification technique. After listing related results, in Section 2 we describe the data set used in this paper. In Section 3 we describe our classification framework. The results of the classification experiments over DC2010 can be found in Section 4.

1.1 Motivation

Classification for trust, bias and neutrality turned out to be very hard with AUC values near 0.5 for the ECML/PKDD Discovery Challenge 2010 participants. Since these attributes constitute key aspects of Web quality, our goal is to improve the classification techniques for these tasks. We concentrate on host level classification, a task suitable for an archivist or researcher compiling a large collection of quality content for further analysis.

As the bag of words representation turned out to describe Web hosts best for most classification tasks of the Discovery Challenge [14], we realized that new text classification methods are needed particularly suited to the quality related tasks in question. Such classifiers are however computationally expensive including SVM that is generally considered to work well for text classification. For example, our previous result [14] uses random forest, a suboptimal choice due to feasibility reasons. Hence we seek for alternative methods that enable expensive classifiers at the Web scale. As one particular technique that is however insufficient in itself, the importance of feature selection already pointed out by DC2010 participants [23].

In order to both improve the quality and reduce the size of the problem, our motivation comes from image processing. Just as Web hosts consist of a collection of individual pages represented by their bags of words, images consist of regions or points of interest represented by high-dimensional image descriptors. Best performing content based image classification systems are typically based on the idea of soft clustering the set of regions and representing images by cluster histograms [25], also called bags of "visual words". The procedure, at the same time, reduces the size of the problem and removes the noise induced by individual outlier image regions.

Based on the above image classification motivation, we bicluster the host-term matrix in order to represent hosts by bags of concepts instead of words. As an additional advantage, the sparse bag of words data is turned to a dense continuous representation of cluster distances, a type of data best suited for SVM classifiers.

Finally for the SVM classification step, after our first discouraging experiments with the document similarity kernel over the distance matrix produced, we considered kernel selection methods [26] that also performed well for the above image classification task [11]. Since the dimensionality of the data is low and it turned out that the performance of various kernels can be measured over a small heldout set, we were able to perform a full comparison of kernel fusion methods from [26].

1.2 Related Results

Closest to the problem of host level quality classification is Web spam filtering, the area of devising methods to identify useless Web content with the sole purpose of manipulating search engine results. Web spam filtering has drawn much attention in the past years [27, 21, 19]. In the area of the so-called Adversarial Information Retrieval workshop series ran for five years [15] and evaluation campaigns, the Web Spam Challenges [3] were organized. The ECML/PKDD Discovery Challenge 2010 (see Section 2) extended the scope by introducing labels for genre and quality by serving the needs of a fictional archive.

Our baseline classification procedures are collected by analyzing the results of the Web Spam Challenges and the ECML/PKDD Discovery Challenge 2010. A key ingredient of the Web Spam Challenge 2008 best result [17] was ensemble undersampling [8] while for earlier challenges, best performances were achieved by a semi-supervised version of SVM [1] and text compression [9]. Best results either used bag of words vectors or the so-called "public" feature sets of [4].

The Discovery Challenge 2010 best result [23] achieved an AUC of 0.62 for non-neutrality, 0.53 for bias and 0.506 for distrust classification while the overall winner [18] was able to classify a number of quality components at an average AUC of 0.80 but their results were below 0.52 for all these categories. As for the technologies, bag of words representation variants proved to be very strong for the English collection. For classification techniques, a wide selection including decision trees, random forest, SVM, class-feature-centroid, boosting, bagging and oversampling in addition to feature selection (Fisher, Wilcoxon, Information Gain) were used [18, 2, 23]. Note that the findings of the usability of feature selection in case of high class imbalance in [23] is similar to our present work. In our previous work [14] we improved over the best results of the Challenge participants; the best performing ingredient of our classifier ensemble was a random forest classifier over a BM25 weighted bag of words representation of the hosts.

We use no link based features; such results are included in [14] that we use as another baseline. One reason is high computational cost. As another reason, our recent classification experiments over DC2010 [14] indicate little use of these features. As a possible reason, the DC2010 training and test set was constructed by handling of hosts from the same domain and IP. Since no IP and domain was allowed to be split between training and testing, we might have to reconsider the applicability of propagation [20, 28] and graph stacking [22]. The Web Spam Challenge data sets were labeled by uniform random sampling and graph stacking appeared to be efficient in several results [5].

2. THE DATA SET

In this paper we use the DC2010 data set created for the ECML/PKDD Discovery Challenge 2010 on Web Quality. The data set is described well in [14], we only summarize

the important aspects with respect to trust and bias.

DC2010 is a large collection of annotated Web hosts labeled by the Hungarian Academy of Sciences (English documents), Internet Memory Foundation (French) and L3S Hannover (German). The base data is a set of 23M pages in 190K hosts in the `.eu` domain crawled by the Internet Memory Foundation early 2010.

The manually created labels included assessment for genre and quality. The motivation behind the labeling procedure was the needs of a fictional Internet archive who may or may not want to completely exclude spam but may prefer certain type of content such as News-Editorial and Educational beyond Commercial sites. Also they may give higher priority to trusted, factual and unbiased content that combine to a utility score.

The DC2010 data set includes hosts labeled by several attributes, out of which spam, trustworthiness, factuality, bias and five genre was selected to be used for classification. While no further labeling is made for a spam host, other properties and in particular the five genre Editorial, Commercial, Educational, Discussion and Personal are non-exclusive and hence define nine binary classification problems. We consider no multi-class tasks in this paper.

Next we summarize assessor instructions concentrating on the three labels relevant for our present work. First, assessors were instructed to check some obvious reasons why the host may not be included in the sample at all, including adult, mixed, language misclassified sites, and then to assess spam. These hosts were skipped for the remaining steps and in particular spam has no bias or trust label.

Hosts were labeled by genre into five categories, news/editorial, commercial, educational, discussion and personal. Important is that discussion spaces are not assessed for bias, i.e., just as spam, skipped for both training and testing. Discussion spaces include dedicated forums, chat spaces, blogs, etc., but comment forms were excluded. We also introduced the distinct category of Personal/Leisure covering arts, music, home, family, kids, games, horoscopes etc. A personal blog for example belongs both here and to "discussion" (and hence not labeled for bias).

Finally, general properties related to trust, bias and factuality were labeled along three scales:

1. Trustworthiness: I do not trust this—there are aspects of the site that make me distrust this source. I trust this marginally—looks like an authoritative source but its ownership is unclear. I trust this fully—this is a famous authoritative source (a famous newspaper, company, organization).
2. Neutrality: Facts—I think these are mostly facts. Fact & Opinion—I think these are opinions and facts; facts are included in the site or referenced from external sources. Opinion—I think this is mostly an opinion that may or may not be supported by facts, but little or no facts are included or referenced.
3. Bias: We adapted the definition from Wikipedia[1]. We flagged flame, assaults, dishonest opinion without reference to facts.

The distribution of labels is given in Table 1. For Neutrality and Trust the strong negative categories have low frequency and hence we fused them with the intermediate negative (maybe) category for the training and testing labels.

[1] `http://en.wikipedia.org/wiki/NPOV`

Label	Yes	Maybe	No
Spam	423		4 982
News/Editorial	191		4 791
Commercial	2 064		2 918
Educational	1 791		3 191
Discussion	259		4 724
Personal-Leisure	1 118		3 864
Non-Neutrality	19	216	3 778
Bias	62		3 880
Dis-Trustworthiness	26	201	3 786
Confidence	4 933		49
Media	74		4 908
Database	185		4 797
Readability-Visual	37		4 945
Readability-Language	4		4 978

Table 1: Distribution of assessor labels in the DC2010 data set.

We also remark that the assessors introduced subjectivity for judging trust: German assessors gave the intermediate maybe as default and yes, no only occasionally. For other assessors, yes was the default value as indicated in Table 1. We use the labels of English sites only and thus avoid this bias.

3. CLASSIFICATION FRAMEWORK

Our first step is to compile bags of concepts from words via biclustering and represent hosts by distances from host cluster centers (Section 3.1). By turning bags of words into cluster distance vectors, we also reduce dimensionality. The effect is reminiscent of latent semantic analysis but for our particular task biclustering seems to fit very well.

Our low dimensional representation allows computationally costly classifiers, in particular SVM, to be applied. Important to note that unlike in the original method [13] that uses Kullback-Leibler divergence, we use Jensen-Shannon, the symmetric version in the biclustering algorithm that makes very large difference in classification quality.

We use a simple supervised feature selection and weighting method based on frequencies in the training set. Unlike for all other categories of spam and genre, this method greatly improves the highly imbalanced classes of non-neutrality, bias and distrust.

Given the compact representation of hosts by cluster distances, we may apply computationally expensive methods for classification. We use SVM with several kernels and apply late fusion as described in Section 3.2. We use libSVM [6].

3.1 Biclustering

Biclustering is a bidirectional clustering algorithm which clusters both the Web hosts and the terms at the same time. The goal is to improve the quality of the clustering by using the clustering along the other axis. In other words biclustering explores a deeper connection between instances and attributes than the usual one-directional clustering methods.

Our biclustering method is based on Dhillon's information theoretic co-clustering algorithm [13]. The basic idea is to consider the data as a joint distribution and maximize the mutual information of row and column clusters.

We applied Dhillon's algorithm with one modification: we

substituted Kullback-Leibler divergence with Jensen-Shannon, its symmetric version. By our experience over several other data sets, this slight modification greatly improves cluster quality.

In our baseline term selection method, we selected the most frequent terms as vocabulary. For efficiency considerations, we selected the top 25,000 terms, even lower than in the DC2010 official data. As in our previous experiments [14], this size constitutes a good compromise between quality and scalability.

Since Dhillon's [13] method is based on information theoretic distances, the raw tf values give best performance for biclustering. Normalized versions such as tf.idf or the BM25 weighting scheme performs significantly worse and is omitted for further consideration.

We applied a simple supervised feature weighting over the same 25000 size vocabulary by selecting terms with increased frequency in the positive instances of the training set. This simple idea results in large gains for our three categories of interest while negligible improvement or even deterioration for spam and genre. One reason could be the the rarity of positive instances in these categories, even lower than in spam. Another reason could be that non-neutrality, bias and distrust depends on special, less frequent terms since these concepts are mostly independent of genre.

For feature selection we computed the ratio of overall tf and the tf of the positive instances. We selected terms where the ratio was below 10. Note that due to the high imbalance this results in strong filtering for category-specific terms. For these terms, we used the category tf as the new weight and united the terms for all categories. Whenever one term is selected for more than one categories, we choose the lower weight. Comparison with more refined selection methods as well as measuring the effect of various parameters and choices in this method is left for future work.

We computed a 500 document times 1000 term class biclustering. We used 20 iterations to typically reach a level of cluster weight changes below 1%. Notice that we reduced our (near) 25,000 dimensions to a mere 1000, hence the possibilities for choosing classifiers is no longer restricted by scalability as it is in the initial bag of words representation. Given the 500×1000 matrix, we performed the following steps to assign a 1000 dimensional vector for every host.

1. Based on the training set, for each cluster and for each category we evaluated the probability that the given cluster belongs to the given category.

2. Based on the similarity of instances to clusters, for every test instance and for every cluster we computed the probability that given instance belongs to the given cluster.

3. By multiplying the above two matrices we get the probability that a test instance belongs to a given category.

The motivation behind biclustering is to group terms into word clusters and hence represent Web hosts as bag of concepts instead of words. The method is very similar to image classification [25] where the so-called bag of visual words representation is created as soft clusters of the low level image elements. In our experience the term clusters carry clear meaning as summarized in Table 2. We note that in addition we found quite a few high weight single-word or few word clusters including ebay, image, friend, lifestyle etc.

Table 3: Kernel functions and parameters.

	Kernel function
linear	$K(x,y) = x' * y$
polynomial	$K(x,y) = (\frac{1}{D}x' * y)^d$
radial basis function	$K(x,y) = e^{(-\gamma(x-y)^2)}$

3.2 Kernel methods

Learning SVM models on text based bag of word models is a widely used technique. One of the main problems is the choice of the well performing kernel functions. The selection or aggregation of basic kernels is typically computationally expensive.

We used a wide range of basic kernels over both the original term and the cluster distance vectors. Our kernels include linear, polynomial and radial basis function with different parameters as seen in Table 3 with D denoting the number of features, $d \in \{1, 2, 4\}$, and $\gamma = \frac{1}{|T|}$ where T is the training set. For kernel combination we applied various cost parameters for the linear kernel.

To determine the final prediction, independently for each category, we set aside 5% of the training as heldout set and tested three strategies:

1. Select best: the best performing model on the heldout.

2. Early aggregation: we combine the kernels according to the ideal weight over the heldout, and let the final prediction for test instance x be

$$pred_{early}(x) = \sum_{i=1}^{N} \alpha_i \sum_{k=1}^{K} \beta_k K_k(x, y_i) + b$$

where $K_k(x, y)$ is the kth kernel function and b is the bias.

3. Late fusion: we combine the SVM outputs according to the ideal weight over the heldout, the final prediction for test instance x is

$$pred_{late}(x) = \sum_{k=1}^{K} \beta_k (\sum_{i=1}^{N} \alpha_{ik} K_k(x, y_i) + b_k)$$

where $K_k(x, y)$ is the kth kernel function and b_k is the bias for the kth SVM classifier.

Because of the low heldout size, the combination weights can easily be determined by setting the weight for the best basic model to 1 and iteratively increase other weights from 0 by steps of 0.1.

3.3 Evaluation metrics

The standard evaluation metrics since Web Spam Challenges [3] is the area under the ROC curve (AUC) [16]. The ECML/PKDD Discovery Challenge used Normalized Discounted Cumulative Gain (NDCG) for evaluation since some tasks used multi-level utility based on spamicity, genre and other attributes. For the binary classification problems we use 1 for a "yes", 0 for a "no" label as utility. These measures perform very similar, even numerically [14]. We describe the version of NDCG applied for DC2010.

Table 2: Example term clusters found by our biclustering algorithm.

yorkie adorable puppy teacup capuchin affectionate akc parrots maltese puppies lovely cute
serbia croatia bosnia albania montenegro macedonia herzegovina belarus moldova kosovo azerbaijan slovak balkans estonian
welcome tel fax submit home please mail click contact reserved
plated earrings necklace pendants necklaces bracelets studs jewelry jewellery
google advertising real category
laptops cheap discount buy
yeah awesome folks wondering okay yes nice maybe pretty hello yesterday guys wow guess
tabs erectile erection pfizer impotence generic
shopping enlarge coupon price
cant lol reply thats btw xd alot logged offline dont pm smf

To emphasize performance over the entire list, the discount function is changed from the common definition to be linear

$$1 - i/N \tag{1}$$

where N is the size of the testing set. To justify the discount function, note that an Internet archive that may crawl 50% or even more of all the host seeds they identify and spam may constitute 10-20% of all the hosts. Our final evaluation formula is

$$\begin{aligned} \text{NDCG} &= \frac{\text{DCG}}{\text{Ideal DCG}}, \text{ where} \qquad (2) \\ \text{DCG} &= \sum_{\text{rank}=1}^{N} \text{utility(rank)} \cdot \left(1 - \frac{\text{rank}}{N}\right), \end{aligned}$$

and Ideal DCG is obtained with utility decreasing with rank. We computed NDCG by the appropriate modification of the python script used by the Yahoo! Learning to Rank Challenge 2010 [7]. We also note here that NDCG and AUC produced numerically very close values on the Discovery Challenge binary problems. The reason may be that both measures show certain symmetry over the value 0.5, although the NDCG for an order and its reverse does not necessarily add up to one due to the normalization in NDCG.

4. RESULTS

In this section we describe our various SVM ensemble methods over the traditional bag of words as well as the cluster distance vector representations. We measure the accuracy of various methods and their combinations. The detailed results are in Table 4. Although we concentrate on neutrality, bias and trust, we give results for all Discovery Challenge 2010 categories to give a better comparison of the techniques used.

For training and testing we use the official official DC2010 set as described in Table 1. As it can be seen, these show considerable class imbalance which makes the classification problem harder.

4.1 Baseline

We have collected the best runs from all DC2010 participants in the first row of Table 4. While genre comes from the winners [18], the high imbalance classes including spam but also the last three of key importance for us was treated best by Wilcoxon feature selection [23].

From our previous results [14], we show class by class the best run as well as random forest over the BM25 weighted bag of words representation, the stable well performing method.

4.2 Bicluster versions

As a basic method without using SVM, we may simply take majority votes within clusters. Unexpected by its simplicity, the method works fairly well and we used this method to select the useful range of parameters, including term and host cluster and iteration count in the biclustering algorithm. The corresponding entries in Table 1 are "bicluster" and "weighted bicluster".

4.3 SVM on top of biclustering results

In our first run using SVM, we selected the best kernel over the heldout set as defined in Section 3.2. Bicluster combination ("bic. comb." in Table 1) consists of the selection of the better from the weighted and unweighted bicluster versions based on the sample training set.

4.4 SVM fusion method

We apply late fusion to tf, bicluster and weighted bicluster. Results are the rows prefixed "fusion" in Table 1, in this order.

4.5 Ensembles

For classifier ensembles, we simply averaged the predictions. We combined all results with the BM25 classifier, postfixed BM25 in Table 1.

5. DISCUSSION

As seen from the results in Table 4, biclustering with supervised term weighting alone already produces promising classification results with a moderately large 500×1000 cluster count. In comparison, the accuracy of the unweighted results for quality aspects remain at the level of a random classifier. Note that the results in the corresponding rows "bicluster" and "weighted" are generated simply by the majority vote of the clusters without using sophisticated classifiers.

The use of SVM over the hosts described as a "bag of clusters" already reaches accuracy close to the best, as seen in row "bic. comb.". This method uses a heldout sample to select the best SVM kernel and the better of term weighting strategies.

The results can finally be improved by the late fusion SVM of weighted and unweighted biclustering as well as the combination of these two techniques. Out of the three vectors based on bag of words as well as unweighted and weighted term biclustering, the latter performs the best, slightly improved by the combination of all three methods.

The main distinction between the different classification tasks relies in whether or not a combination with the random

		spam	news	commercial	research education	discussion	personal leisure	(non)neutral	biased	(dis)trusted	**quality average**	***average***
DC2010 best	AUC	0.830	0.734	0.840	0.840	0.777	0.801	0.626	0.558	0.506	0.563	0.723
	NDCG	0.833	0.740	0.883	0.885	0.784	0.828	0.620	0.553	0.510	0.561	0.737
best [14]	AUC	**0.891**	0.808	0.799	0.827	0.850	0.808	0.618	0.653	0.582	0.612	0.754
	NDCG	**0.893**	0.811	0.852	0.875	0.865	0.838	0.624	0.656	0.586	0.617	0.771
BM25	AUC	0.876	0.787	0.779	0.816	0.843	0.797	0.580	0.653	0.520	0.584	0.739
	NDCG	0.879	0.791	0.838	0.868	0.848	0.825	0.587	0.656	0.534	0.589	0.704
bicluster	AUC	0.813	0.706	0.686	0.726	0.640	0.674	0.508	0.476	0.445	0.476	0.631
300x1000	NDCG	0.817	0.711	0.770	0.803	0.653	0.719	0.516	0.481	0.450	0.482	0.657
weighted 500x1000	NDCG	0.817	0.719	0.757	0.814	0.771	0.699	0.512	0.592	0.572	0.558	0.694
bic. comb. SVM	AUC	0.821	0.791	0.866	0.858	0.793	0.832	0.632	0.611	0.634	0.625	0.760
	NDCG	0.825	0.795	0.902	0.898	0.800	0.855	0.638	0.615	0.637	0.630	0.774
fusion tf	AUC	0.732	0.592	0.689	0.718	0.712	0.686	0.558	0.463	0.550	0.523	0.633
	NDCG	0.737	0.600	0.772	0.797	0.722	0.729	0.565	0.468	0.554	0.529	0.661
fusion bicluster	AUC	0.816	0.797	0.858	0.858	0.803	0.833	0.607	0.534	0.623	0.588	0.748
	NDCG	0.819	0.801	0.896	0.898	0.810	0.856	0.614	0.539	0.627	0.593	0.763
fusion weighted	AUC	0.824	0.742	0.862	0.859	0.817	0.826	0.630	0.611	**0.638**	0.626	0.756
	NDCG	0.828	0.747	0.899	0.898	0.824	0.849	0.636	0.615	**0.641**	0.630	0.771
fusion all three	AUC	0.835	0.794	**0.869**	0.858	0.830	0.838	**0.637**	0.611	**0.638**	**0.628**	0.768
	NDCG	0.838	0.798	**0.904**	0.897	0.836	0.860	**0.643**	0.615	**0.641**	0.633	0.781
fusion	AUC	0.874	**0.833**	0.862	0.866	0.862	**0.850**	0.594	0.669	0.566	0.610	0.775
bicluster+BM25	NDCG	0.876	**0.836**	0.899	**0.904**	0.867	**0.870**	0.601	0.673	0.570	0.614	0.789
fusion	AUC	0.882	0.800	0.863	0.864	0.861	0.838	0.622	**0.682**	0.577	0.627	0.777
weighted+BM25	NDCG	0.884	0.804	0.899	0.902	0.866	0.860	0.628	**0.685**	0.581	**0.634**	0.790
fusion	AUC	0.880	0.831	0.864	**0.867**	**0.869**	0.849	0.622	**0.682**	0.577	0.627	**0.782**
bic. comb.+BM25	NDCG	0.883	0.834	0.900	**0.904**	**0.874**	**0.870**	0.628	**0.685**	0.581	**0.634**	**0.795**
fusion all+BM25	AUC	0.882	0.830	0.864	0.865	0.868	0.849	0.605	0.682	0.577	0.621	0.780
	NDCG	0.885	0.833	0.900	0.902	0.872	0.870	0.612	0.685	0.581	0.626	0.793

Table 4: Detailed performance over the DC2010 labels in terms of AUC and NDCG as in equation (2)**.**

forest text classifier over the BM25 term weighting scheme improves the final result. Most genre except commercial as well as bias have relative good BM25 based performance and this is improved by combining with the two biclustering or all three SVM classifiers. Classification for commercial slightly while neutrality and trust are strongly deteriorated when combining with the BM25 based output. Neutrality and trust seem to require classification techniques very different from other aspects.

6. CONCLUSIONS

Over the 190,000 host DC2010 data sets, we gave methods to classify Web hosts for neutrality, bias and trust. We described perhaps the first attempt of a practically useful quality classification with AUC stable above 0.6 by an improved text classification method. By biclustering hosts and the top 25,000 most frequent terms, we reduce the term space to groups of words and also represent hosts by their distances from cluster centers. On top of our new host representation, we use fusion methods in SVM. Surprisingly, unlike for the more traditional tasks as spam or genre classification where our new method gives marginal improvement, if any, for the hard tasks of neutrality, bias and trust we obtain strong improvement over the baseline of existing host-level classification methods. This fact indicates that especially neutrality and trust behaves very different from the well-known tasks of spam and genre classification.

We consider our results as first step with several technologies remaining open to be explored. For example, unlike expected, the ECML/PKDD Discovery Challenge 2010 participants did not deploy natural language processing based features.

Acknowledgment

To the large team of organizers and assessors for the complex labeling process of the DC2010 data set.

7. REFERENCES

[1] J. Abernethy, O. Chapelle, and C. Castillo. WITCH: A New Approach to Web Spam Detection. In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2008.

[2] L. D. Artem Sokolov, Tanguy Urvoy and O. Ricard. Madspam consortium at the ecml/pkdd discovery challenge 2010. In *Proceedings of the ECML/PKDD 2010 Discovery Challenge*, 2010.

[3] C. Castillo, K. Chellapilla, and L. Denoyer. Web spam challenge 2008. In *Proceedings of the 4th International*

Workshop on Adversarial Information Retrieval on the Web (AIRWeb), 2008.

[4] C. Castillo, D. Donato, L. Becchetti, P. Boldi, S. Leonardi, M. Santini, and S. Vigna. A reference collection for web spam. *SIGIR Forum*, 40(2):11–24, December 2006.

[5] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: web spam detection using the web topology. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 423–430, 2007.

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

[7] O. Chapelle, Y. Chang, and T.-Y. Liu. The yahoo! learning to rank challenge, 2010.

[8] N. Chawla, N. Japkowicz, and A. Kotcz. Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6, 2004.

[9] G. Cormack. Content-based Web Spam Detection. In *Proceedings of the 3rd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2007.

[10] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20, 1995.

[11] B. Daróczy, A. Benczúr, and R. Pethes. SZTAKI at ImageCLEF 2011. In *Working notes of CLEF 2011*, 2011.

[12] K. Dave, S. Lawrence, and D. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM, 2003.

[13] I. Dhillon, S. Mallela, and D. Modha. Information-theoretic co-clustering. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 89–98, 2003.

[14] M. Erdélyi, A. Garzó, and A. A. Benczúr. Web spam classification: a few features worth more. In *Joint WICOW/AIRWeb Workshop on Web Quality (WebQuality 2011) In conjunction with the 20th International World Wide Web Conference in Hyderabad, India.* ACM Press, 2011.

[15] D. Fetterly and Z. Gyöngyi. Fifth international workshop on adversarial information retrieval on the web (AIRWeb 2009). 2009.

[16] J. Fogarty, R. S. Baker, and S. E. Hudson. Case studies in the use of roc curve analysis for sensor-based estimates in human computer interaction. In *Proceedings of Graphics Interface 2005*, GI '05, pages 129–136, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.

[17] G. Geng, X. Jin, and C. Wang. CASIA at WSC2008. In *Proceedings of the 4th International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2008.

[18] X.-C. Z. Guang-Gang Geng, Xiao-Bo Jin and D. Zhang. Evaluating web content quality via multi-scale features. In *Proceedings of the ECML/PKDD 2010 Discovery Challenge*, 2010.

[19] Z. Gyöngyi and H. Garcia-Molina. Spam: It's not just for inboxes anymore. *IEEE Computer Magazine*, 38(10):28–34, October 2005.

[20] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, pages 576–587, Toronto, Canada, 2004.

[21] M. R. Henzinger, R. Motwani, and C. Silverstein. Challenges in web search engines. *SIGIR Forum*, 36(2):11–22, 2002.

[22] Z. Kou and W. W. Cohen. Stacked graphical models for efficient inference in markov random fields. In *SDM 07*, 2007.

[23] V. Nikulin. Web-mining with wilcoxon-based feature selection, ensembling and multiple binary classifiers. In *Proceedings of the ECML/PKDD 2010 Discovery Challenge*, 2010.

[24] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[25] F. Perronnin, C. Dance, G. Csurka, and M. Bressan. Adapted vocabularies for generic visual categorization. In *Computer Vision–ECCV 2006*, pages 464–475, 2006.

[26] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. simplemkl. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

[27] A. Singhal. Challenges in running a commercial search engine. In *IBM Search and Collaboration Seminar 2004*. IBM Haifa Labs, 2004.

[28] B. Wu, V. Goel, and B. D. Davison. Topical TrustRank: Using topicality to combat web spam. In *Proceedings of the 15th International World Wide Web Conference (WWW)*, Edinburgh, Scotland, 2006.