2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology

Leveraging Web 2.0 Sources for Web Content Classification

Somnath Banerjee
Hewlett-Packard Labs
Bangalore, India
somnath.banerjee@hp.com

Martin Scholz
Hewlett-Packard Labs
Palo Alto, CA, USA
scholz@hp.com

Abstract

This paper addresses practical aspects of web page classification not captured by the classical text mining framework. Classifiers are supposed to perform well on a broad variety of pages. We argue that constructing training corpora is a bottleneck for building such classifiers, and that care has to be taken if the goal is to generalize to previously unseen kinds of pages on the web. We study techniques for building training corpora automatically from publicly available web resources, quantify the discrepancy between them, and demonstrate that encouraging agreement between classifiers given such diverse sources drastically outperforms methods that ignore the different natures of data sources on the web.

1. Introduction

The classification of textual documents is a typical and important machine learning task with an enormous variety of applications. Web related examples where such classifiers are successfully applied include the categorization of news and blog content, spam filtering, and the filtering of web content with respect to user-specific interests.

The classical text classification literature focuses on controlled clean corpora, single-label problems, and most noticeably, it comes with the *i.i.d. assumption*, that is, the assumption that training and test set were sampled from the same underlying distribution. This assumption simplifies matters a lot, because in this case cross-validation results are reliable indicators for how well the target concept has been learned. We claim that many web page classification tasks do not fit this classical setting of classification. Often the goal is to train classifiers that perform well on a broad variety of pages, ranging from clean dictionary entries over Internet shopping sites to noisy blogs. We cannot expect to get an i.i.d. sample from the target distribution, because it may not even be a fixed distribution but depend on the current user, and even the labels may sometimes be debatable. Another reason in practice is that a procedure of sampling from the "right" distribution would involve crawling representative sites from the web and manually labeling them, which is too tedious and expensive for many tasks. Constructing labeled training corpora is a major bottleneck for supervised text mining applications, which e.g., gave rise to semi-supervised techniques like multi-view learning [22] that aim for minimizing the demand for labeled data.

Just recently, part of the research community started to work on various practically relevant, but much more difficult classification settings in which the classifiers are used with data distributions that do not obey the convenient i.i.d. rule at deployment time. Examples include learning under concept drift (e.g., [8]), where the target concept changes over time, transfer learning (e.g., [1]) where classifiers learned for similar but different concepts are exploited to learn the target concept, and learning under sample selection bias [6], which subsumes a variety of other problems that involve non-i.i.d. sampled training sets.

In this paper, we are addressing practical aspects of web page classification: the construction of training data sets for broadly applicable multi-label web page classifiers. To this end, we propose a general framework that involves gathering and labeling data in an automatic fashion by utilizing various sources on the web. This approach avoids both the process of manually labeling web pages and the complexity of semi-supervised learning, while promising to produce highly accurate classifiers.

The contribution of this paper is twofold. First, we demonstrate that it is possible to build accurate web page classifiers without the hassle of labeling examples manually, while avoiding the pitfalls of unrealistic assumptions like stationary and homogeneous data collections that dominate the literature. Second, we show that the different web sources we exemplary use in this paper in fact follow different underlying distributions, and that the diversity found in these sources cannot be ignored. Finally, we study which sources and combination strategies are able to overcome the discovered problems, and hence allow to build the widely applicable classifiers we aim for.

978-0-7695-3496-1/08 $25.00 © 2008 IEEE
DOI 10.1109/WIIAT.2008.291

The remainder of this paper is organized as follows: In the next section we discuss the construction of training corpora from exemplary chosen publicly available sources on the web. In Section 3, we describe the experimental setup that we use in Section 4 to evaluate and analyze our baseline classifiers. In response to the results, we propose and evaluate different strategies in Section 5. Section 6 concludes.

2. Building training sets automatically

The Web 2.0 provides a variety of resources that are promising for data mining problems. It leveraged joint efforts like tagging web contents and building up structured and semi-structured knowledge in electronic form, most prominently the online encyclopedia Wikipedia[1] and the open directory of web pages, DMOZ[2]. The results are manually labeled, structured, or annotated, and – as we will show in the remainder of this paper – can be used as cheap proxies for self-labeled data.

Our goal is to construct training sets for real-world web page classification. It is often problematic and inappropriate to assume that all web pages exclusively belong to a single category. We hence frame our learning problem as a set of binary classification problems, in which each document can belong to none, one, a few, or all categories. A binary classifier determines the estimated membership function for each category. Each training corpus (labeled training set from a specific source for a specific category) hence contains positive and negative examples. For brevity, our studies focus on a representative set of four sources and corresponding extraction and labeling techniques. Further sources and different extraction methods can be integrated easily. The selection of sources will be motivated inline. Our extraction techniques were all chosen to be (i) simple enough to be reproducible, (ii) generic enough to cover a vast majority of potentially relevant concepts, and (iii) not to require extensive human interaction during the corpus construction phase. As we shall see, they still allow to build highly accurate classifiers. The specific problems (categories) this paper exemplary focuses on will be described in more detail in Section 3.1.

Open directory (DMOZ). DMOZ is a human edited web directory that contains almost 5 million web pages, categorized under nearly $600,000$ categories. Each category in DMOZ represents a concept, and the categories are organized hierarchically.

The way DMOZ structures the data in terms of natural, human interpretable concepts and the fact that every page is interpreted and classified by a human annotator makes the DMOZ collection the most natural and probably most popular choice for building training sets in the web domain (e.g., [5]). It basically provides a gigantic, manually labeled training set which could roughly reflect the distribution underlying the WWW.

For our experiments, we crawled an RDF dump of DMOZ from November 26, 2006, and we downloaded all pages referenced in that dump. Pages in the sub-tree rooted at any specific category can be thought of as the positive examples of the corresponding class, and the remaining pages as negatives.

We constructed training corpora from DMOZ by selecting 1000 positive examples by "breadth first search" in the corresponding sub-trees of relevant categories, and an equal amount of negative examples chosen at random from pages outside those trees.

Search engine (Google). Search engines provide a simple interface to obtain web pages for any given concept. We simply used the target concept name (e.g., *photography*) as a search query to Google's search engine[3] and used the landing pages of the first 1000 hits as positive examples; we selected 1000 negatives from DMOZ in the same way as described above.

When considering search results as training examples, one should keep in mind that the pages are relevant in terms of e.g., the PageRank [2] measure, but that they do not necessarily provide definitions for the queried term(s) or any kind of descriptive content. Many of the pages hence have a low signal. For example, start pages of topic-specific portals are legitimate search results, but they often contain more ads and navigational parts than descriptive text.

However, search engines have recently been recognized as useful for gathering data sets in different contexts, e.g., n-gram statistics [14] and meaning discovery [3]. Fergus et al. [7] used the image search engine of Google to automatically gather training examples for object category recognition, and most recently a bootstrapping scheme has been suggested for text classification [11].

Social bookmarking site (Del.icio.us). Del.icio.us[4] is a social bookmarking site that allows users to save and tag URLs. The tags used by multiple people for a particular URL are often quite representative of the concept mentioned in the web page of the URL. Pages tagged with a concept name can be thought of as the positive examples for that concept. Tagging is a very recent activity on the web, so not many text mining approaches utilize tags so far. One example is Yanbe et al. [21], who showed that tags can be used to improve the results of web search engines. An

[1] http://www.wikipedia.org/
[2] http://www.dmoz.org/
[3] http://www.google.com
[4] http://del.icio.us/

advantage of social tagging in our setting is that tags capture semantics in a way that resembles human perception at an appropriate level of abstraction, without introducing any unnatural assumptions, like categories being mutually exclusive. The tags are unstructured and freely composable.

Del.icio.us provides an API to obtain web pages with any specified tag. For a category *photography* we would simply use the Del.icio.us API to obtain pages tagged with the term "photography". Positive examples for Del.icio.us are obtained by crawling 1000 (wherever available) such pages. An equal number of negative examples from DMOZ are chosen as outlined above.

Encyclopedia (Wikipedia). Wikipedia is a community edited encyclopedia containing pages in many different languages. The English version of Wikipedia contains 2+ million pages and has 7+ million registered users[5]. A recent study states that the coverage as well as the quality of Wikipedia is comparable to encyclopedia Britannica [10].

Important properties of Wikipedia in our context include (i) its semi-structured nature, with no labels being given a priori, but therefore (ii) deeper semantics when comparing to any of the other considered sources, and (iii) very clean pages that provide definitions and refer to related concepts. Wikipedia was recently successfully utilized for various text mining applications, including text categorization [9, 19], text clustering [13], named entity disambiguation [4], and improving search results [16].

To gather positive examples for a given concept, we constructed a Lucene[6] index of the Wikipedia dump and used the target concept as the search query, in the same way as described for the Google corpus. Again, we used the top 1000 pages as our positive examples. For selecting negative examples for a concept, we excluded the first 2000 hits (returned by Lucene) from Wikipedia for each query under consideration, and then sampled 1000 negative examples from the remaining pages. We found that it is necessary to also use Wikipedia pages rather than DMOZ pages as negatives, because pages from Wikipedia and DMOZ have quite different characteristics.

We also tried graph-based page similarity calculation techniques to retrieve similar pages for a given concept. As long as the given concept can be characterized by a specific Wikipedia page, there are random walk techniques to obtain similar pages. We experimented with *Topic Sensitive PageRank* [12] and *Green Measure* [17], which both did not work well. The top few results were usually good, but the noise level (unrelated pages) rapidly increased when going further down the list, so retrieving 1000 relevant pages for our training corpus was not possible with these techniques.

[5] `http://en.wikipedia.org/wiki/Special:Statistics`
[6] `http://lucene.apache.org/`

3. Experimental Setup

The previous section described ways of obtaining training examples from different web sources for almost any given category. This data can be used to train a set of binary classifiers in the next step. This section describes our test bed and justifies our evaluation scheme, before we move on to the actual experiments in the following section.

3.1. Data Sets for Evaluation

We selected a set of 10 diverse concepts for our empirical evaluation. For the sake of simplicity and clarity, our concepts were chosen as to satisfy the following constraint: Each concept had to match a category name in DMOZ and a tag in Del.icio.us. This does not narrow down applicability in practice; if there is no exact match then a number of very similar categories/tags can easily be substituted.

We used the concepts *health*, *shopping*, *science*, *programming*, *photography*, *linux*, *recipes*, *web design*, *humor* and *music*, which span across three different levels of the DMOZ hierarchy and therefore vary considerably in terms of specificity. For each of these 10 concepts, we constructed a separate training corpus from each of the four different sources. Each corpus contained 1000 positive examples and an equal number of negatives as discussed in Section 2.

We preprocessed the raw HTML pages by removing any non-textual content (HTML tags and scripts), tokenized the page, removed stop words, and applied a Porter stemmer. We removed all pages that contained less than 50 words at that point, because they usually did not refer to the concept under consideration. For the experiments, we finally applied the standard TF-IDF weighting and built binary classifiers for each concept using the SMO-SVM of the Weka library [20] with the default settings. Our evaluation measure is classification accuracy averaged over all categories, which is similar to F-Measure in our case, because our corpora have balanced class distributions.

3.2. Evaluation strategy

In the common data mining setup, we would simply cross-validate our learning algorithm on the data sets mentioned above. We do not assume the training corpus to resemble the distribution of web pages at deployment time, however. We will hence only use cross-validation in the specific case where we evaluate classifiers on the same source it was trained on. For the most part, we will switch to evaluation schemes where we evaluate on a single source that is not available during training, and we will compare different strategies for constructing well suited training sets from the remaining (three) sources in this setting.

Our rationale is that each corpus will typically contain noise and systematic mistakes. The DMOZ concept of *photography* does not subsume *underwater photography*, for example. The mere fact that a DMOZ category name, a Del.icio.us tag, and a Wikipedia page title are identical does not necessarily imply identical underlying semantics.

We assume that there is still agreement between large parts of the different taxonomies, tags, and labels, respectively. This agreement is rooted in a common conceptualization of human annotators. The degree of agreement can be interpreted as compatibility when using different corpora in the same experiment. Similar to transfer learning, having learned a DMOZ concept will give us a good prior for the same concept defined by e.g., Del.icio.us instead.

Our primary goal is hence to find techniques that allow to learn good classifiers for each source, even if that source is not part of the training set. This is similar to hold-out evaluation, but at the level of data sources. In order to capture and quantify the difference between individual corpora, as well as the generality of a classifier learned under specific circumstances, we will also use another evaluation strategy that applies to multiple training corpora with different underlying distributions. The evaluation of a classifier trained under a different distribution than the one it is deployed on is known from various settings, e.g. classification under sample selection bias [6]. In our scope, we refer to this strategy as *cross-corpus evaluation*. We evaluate ordered pairs (i, j) of corpora by training a classifier on corpus i and then measuring its performance on corpus j. For any pair of corpora that share a common underlying distribution, the expected cross-validation and cross-corpus evaluation results would be identical, whereas, for any pair of highly incompatible corpora applying each of the classifiers to the other corpus would result in much lower classification accuracies.

4. Experiments with a Baseline Method

This section presents empirical results for the baseline method that simply ignores that our web corpora might all have different characteristics. It just merges the data of the three corpora that are available for training and fits an SVM classifier to that single training data set.

To have a reference point for evaluating the different methods, we will first discuss the results of the corresponding cross-corpus evaluation. Figure 1 shows an overview of all the pairwise performances. For each pair of category and data source we trained a separate classifier and applied it to all data sets of the same category. We substituted tenfold cross-validation results whenever the same source was used for training and testing. We will (for the most part) discuss aggregates, i.e., accuracies averaged over all of our 10 concepts in this paper. Table 1 shows such aggregates over all results where the classifier was built from the same source (row) and applied to another source (column). Again, all entries on the main diagonal are averages of the corresponding 10-fold cross-validation accuracies.

	Google	Delicious	DMOZ	Wikipedia
Google	(96.44)	**84.17**	*63.42*	**87.12**
Delicious	**90.00**	(93.54)	**68.36**	76.71
DMOZ	*79.98*	77.15	(83.92)	*75.84*
Wikipedia	88.28	*76.21*	65.05	(94.26)

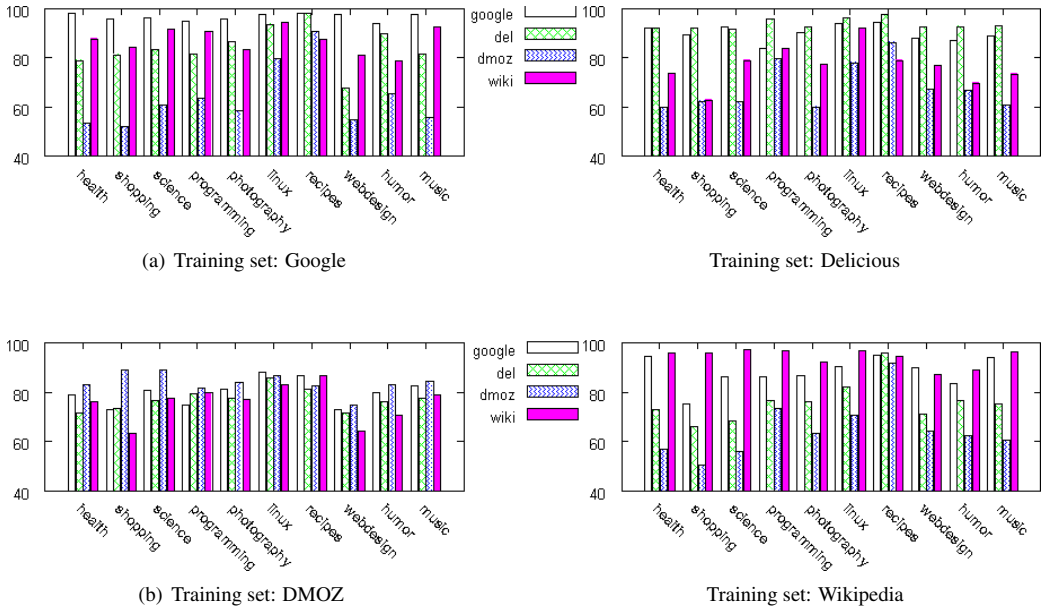
Table 1. Results of cross-corpus evaluation. Rows are training sets, columns the test sets.

For convenience, we set the highest accuracy achieved by any other corpus for each test corpus in bold (maximum of each column) and the lowest accuracy in italics. The numbers in bold are reasonable lower-bounds for how much of the "real" concept is reflected on average by the corpus; it shows the agreement with a classifier built from an independent data set that is only connected to the training set via a common concept name used for both the corpus constructions. The 10-fold cross-validation accuracies (main diagonal) can be referred to as the corresponding upper-bounds of the accuracy we can hope for when given a sample of the same size, because the error rate under these (effectively) i.i.d. samples is an artifact of the learning strategy and not caused by a difference between train and test distribution.

The second line in Table 2 shows the results of the baseline method that simply aggregates the training corpora. For each category we combined the three different training sources with equal weights and tested on the remaining corpora (column). For example, when a single training corpus is created by combining the data (positive and negative) of Del.icio.us, DMOZ and Wikipedia for each concept, it gives 76.94% accuracy on average on the 10 Google corpora. Note that the training sets in this experiment are 3 times larger than in the cross-corpus matrix. Since we do not assume to know the test set at training time, we aim for a strategy that gives us performances that are close to the bold numbers in Table 1, repeated in line 1 of Table 2. In this light, the performance of the baseline strategy is very poor. Surprisingly, in 3 out of 4 cases this method performs just about as good as the worst single-source classifier, see Table 1. When testing on DMOZ, the result is in an acceptable range, but low in absolute terms.

Looking at Figure 1 and inspecting some pages from the different sources confirms the assumptions made in Section 3.2; the corpora differ in fact considerably, and mixing them blindly results in far noisier, heterogeneous, and non-separable corpora that contain examples that sometimes systematically contradict each other because of different concept definitions used by different sources.

An interesting finding is the poor performance of the



(a) Training set: Google

Training set: Delicious

(b) Training set: DMOZ

Training set: Wikipedia

Figure 1. Pairwise cross-corpus evaluation results at the category level. Each of the four blocks describes the result for a common training set. The colors of the bars indicate the test sets.

community-built DMOZ collection of web pages in our experiments, compared to the much simpler Google corpora. We think this contradicts common beliefs, and that the problems we found illustrate why corpora systematically contradict each other for some concepts: The DMOZ categories are not organized in terms of a taxonomy (tree), but of a directed graph. Care has to be taken when constructing the sub-tree for a concept, so that all the positive examples of the concept are covered and cannot be sampled as negatives. This is hard, however, because a large number of categories in DMOZ are spread out over the concept graph without any proper path connecting the pieces. For example, the top level categories *Regional*, *Reference*, and *News* cover many categories that reoccur in other parts of DMOZ without any connecting path. In turn, there are debatable links at the level of the concept hierarchy that affect large sub-trees of documents in a systematic way, i.e., they might become false positives. This does not compromise the cross-validation accuracy, because in this case the evaluation happens on uniform sub-samples of the same corpus, so the classifier might capture all these conventions quite well. However, the classifier's concept may not reflect the natural meaning very well, and consequently does not generalize to other corpora. This is why cross-corpus validation is a useful tool. We addressed noise issues during corpus construction, e.g. by respecting the complex link structure and by disregarding known-noisy nodes; still we expect our corpora to contain unreliable labels that hurt performance.

5. Leveraging multiple sources

In response to the bad performance of the baseline method and the demonstrated differences between the distributions underlying different web sources we will evaluate two different strategies in this section. The first one is an ensemble technique, the second one refines the baseline method by introducing example weights before training.

5.1. Majority Vote

Please recall that for predicting the labels of an (assumed unknown) corpus we have three training corpora available in our experimental setup. Figure 1 shows that among those corpora there usually is at least one that would work very well compared to the baseline method. For this reason, we did not mix training data from different sources for the experiments in this subsection, but trained a separate classifier for each training corpus. We applied Platt's scaling [18] to turn SVM outputs into calibrated probability estimates.

As our remaining challenge, it is generally unknown which training set provides good performance on a previously unseen example. The predictive performance of training sets apparently depends on both the concept and the test set. A natural goal is to get an overall predictive performance that is close to the best individual classifier. This goal is known as tracking the best expert in the literature on concept drift [15], where the term *expert* refers to individ-

ual classifiers. If true labels were revealed after classifying each instance then we were in the setting of classifying data streams under concept drift.

Instead, we address the harder, but more relevant case, in which the true labels are not revealed, so we cannot dynamically adapt classifier weights. In this case, we can still pick appropriate weights offline. Line 3 in Table 2 shows the results of an unweighted majority vote, where we averaged the soft predictions of three classifiers and tested on the indicated (column header) fourth corpus of that category.

It can be seen that – by just keeping the corpora separate during training – we already improved drastically, and got results that are (on average) much closer to the best available individual classifier shown in the first line of Table 2. Under our weak assumptions, this implies that these classifiers generalize better to unseen corpora (or different types of web pages). Weighting the classifiers, e.g., by their average cross-corpus performance did not further improve the results in our experiments, so we skip the details.

5.2. Weighting training data by confidence

The second strategy is to encourage agreement between the different views on the same concept. It is inspired by multi-view learning, see e.g. [22], a semi-supervised technique where each data point has multiple representations. For multi-view learning techniques, the generalization error can be upper-bounded surprisingly well if the learner manages to enforce agreement on both the labeled and unlabeled data across the different views. The theory requires unlabeled data and test set to follow the same distribution.

Our setting is more complex, since we do not even have an unlabeled sample from the target distribution. But still we can encourage agreement between the classifiers regarding their *training* sets. Before describing this in more detail, it is worth fleshing out the different reasons why an example e_B sampled from source B could be misclassified by a classifier C_A trained on source A:

1. The example e_B is noisy, that is, the labeling process just failed or someone inserted a bad reference between concepts into DMOZ etc.

2. The abstraction from the data collected from A to the function C_A done by the learner is imperfect. Even all the cross-validation results in Table 1 are below 100%. So the label of e_B is correct, C_A errs.

3. The classifier and the label of the examples are both correct, but the conventions between the two corpora A and B are inconsistent. This is no "noise" in the classical sense, but a different kind of problem, related to transfer learning: Concepts are similar but not identical across different sources.

Given the lack of reliable information, it should be clear that it is very hard to address point 3, if we have to handle points 1 and 2 at the same time. For this reason, our solutions cannot be as grounded theoretically as techniques in multi-view learning, but still intuitively compelling.

We start with the classifiers that we trained in the last subsection. Our advantage over the setting sketched above is that we have multiple classifiers that we can consult in order to decide on the expected utility of e_B, not only a single classifier C_A. This allows us to incorporate the confidence of all the different "views" captured by classifiers trained from different, independent sources, which helps to reduce the impact of any source-specific noise.

To this end, we prepare a weighted version of each category-specific training set. For computing the weight of any specific example e_B, with word vector x_B and label y_B from source B, we exclude the classifiers trained from source B itself. We perform an unweighted majority vote of the remaining classifiers to determine the weights. In more detail, each of our binary classifiers not trained from B gives us a calibrated probability estimate $P(y_B \mid x_B)$ for e_B for each category. We use the average of those estimates as the weight for e_B. As desired, examples hence require the agreement of classifiers trained from different sources in order to receive a high weight. For each of the three training corpora per category, there are two classifiers available to determine the weights in our specific experiments. In the next step, we join all examples from the three training sources to a single corpus per category, similar to Section 4. The difference is that we use the weights during the subsequent step of classifier training. Finally, we apply the classifiers to the test corpora of the fourth source. Note that the test corpora are at no point used during training.

Line 4 in Table 2 shows the averaged accuracies for this weighting technique. Compared to the equal weight combination (line 2), the accuracy improved drastically. It is now comparable to the performance of majority voting. This indicates that corpus-specific noise is the main reason for the bad baseline performance, and that it can be mitigated by seeking agreement between the various sources.

We want to take this idea one step further. Bad conventions, missing or questionable links between categories, and other kinds of white and systematic noise all share the property that they are not found across multiple sources, but are local problems. As a final refinement of the weighting strategy, we exclude all examples for which the majority (here: "both") of classifiers predicts the opposite label when making a discrete (boolean) prediction, and assign the highest possible weight of 1 whenever all classifiers predict the label assigned to the example. In the latter case, all classifiers agree that an example has the correct label. In the former case, the label of the example is very questionable when leaving the context of the specific source it came from. We

	Google	Delicious	DMOZ	Wikipedia
Best single cross-corpus result (4)	90.00	84.17	68.36	87.12
Equal weight combination (4)	76.94	76.47	68.74	76.79
Majority Vote (5.1)	92.95	84.45	65.10	84.44
Weighted training instances (5.2)	93.88	84.65	66.08	85.73
Weighting & noise elimination (5.2)	93.29	87.52	70.21	87.50

Table 2. Results of the different methods discussed in Sections 4 and 5.

conclude that it is not helpful to include it at all, but that we are either in case 1 or in case 3 above. In all other cases, there is no consensus. Most, but not all of our classifiers confirm that the example has the correct label attached. As before, we weight these examples by confidence that their labels are correct in a general scope.

The last line in Table 2 shows the results for this stronger noise reduction and emphasis on agreement. The averaged accuracies are higher for Del.icio.us, DMOZ and Wikipedia compared to just weighting training instances (line 4). In case of the Google data set, the previous results were already quite good ($\approx 93\%$) and we observe no further improvement. Overall, the results of this method are better than all other methods we tried so far. Please note that by utilizing the cross-corpus diversity we were able to outperform the best single-corpus accuracies, see line 1.

6. Conclusions

We showed that it is possible to build highly accurate classifiers from Web 2.0 sources with low efforts and costs. Our evaluation assumed that we do not have any fixed test set, but rather want to find sources that give good results in a variety of different domains. We used cross-corpus evaluation to quantify the discrepancy between corpora constructed from different sources. It turned out that ignoring the diversity of sources is a surprisingly bad strategy. In response, we developed a cross-corpus example selection and weighting technique based on a set of mild assumptions that takes advantage of this diversity. With our novel learning schemes it is possible to build classifier that perform consistently well across different sources.

References

[1] S. Banerjee. Boosting inductive transfer for text classification using wikipedia. In *Proc. Int. Conf. on Machine Learning and Applications (ICMLA)*, pages 148–153. IEEE, 2007.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW Conference*, 1998.

[3] A. Cilibrasi and P. Vitanyi. The google similarity distance. *IEEE Trans. Knowledge and Data Engineering*, 19(3), 2007.

[4] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Empirical Methods in Natural Language Processing*, 2007.

[5] D. Davidov, E. Gabrilovich, and S. Markovitch. Parameterized generation of labeled datasets for text categorization based on a hierarchical directory. In *SIGIR*, 2004.

[6] W. Fan and I. Davidson. Reverse testing: an efficient framework to select amongst classifiers under sample selection bias. In *Knowledge discovery and data mining (KDD)*, 2006.

[7] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *ICCV Conference*, 2005.

[8] G. Forman. Tackling concept drift by temporal inductive transfer. In *SIGIR Conference*, pages 252–259. ACM, 2006.

[9] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*, 2006.

[10] J. Giles. Internet encyclopedias go head to head. *Nature*, 483:900–901, 2005.

[11] R. Guzman-Cabrera, M. M. y Gomez, P. Rosso, and L. Villasenor-Pineda. Improving text classification by web corpora. In *Proc. of the 5th Atlantic Web Intelligence Conference (AWIC)*, 2007.

[12] T. Haveliwala. Topic-sensitive pagerank. In *WWW Conference*, 2002.

[13] J. Hu, L. Fang, Y. Cao, H. J. Zeng, H. Li, Q. Yang, and C. Z. Enhancing text clustering by leveraging wikipedia semantics. In *SIGIR*, 2008.

[14] F. Keller and M. Lapata. Using the web to obtain frequencies for unseen bigrams. *Computational Linguistics*, 29, 2003.

[15] J. Z. Kolter and M. A. Maloof. Using additive expert ensembles to cope with concept drift. In *Proc. of ICML*, 2005.

[16] D. N. Milne, I. H. Witten, and D. M. Nichols. A knowledge-based search engine powered by wikipedia. In *ACM CIKM Conference*, 2007.

[17] Y. Ollivier and P. Senellart. Finding related pages using green measures: An illustration with wikipedia. In *AAAI Conference*, 2007.

[18] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In *Adv. in Large Margin Classifiers*, 1999.

[19] P. Wang, J. Hu, H. J. Zeng, L. Chen, and Z. Chen. Improving text classification by using encyclopedia knowledge. In *ICDM Conference*, 2007.

[20] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[21] Y. Yanbe, A. Jatowt, S. Nakamura, and K. Tanaka. Can social bookmarking enhance search in the web? In *International Conference on Digital Libraries*, 2007.

[22] Z.-H. Zhou and M. Li. Tri-training. exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering*, 17(11):1529–1541, 2005.