Using Links to Classify Wikipedia Pages

Rianne Kaptein[1] and Jaap Kamps[1,2]

[1] Archives and Information Studies, Faculty of Humanities, University of Amsterdam
[2] ISLA, Faculty of Science, University of Amsterdam

Abstract. This paper contains a description of experiments for the 2008 INEX XML-mining track. Our goal for the XML-mining track is to explore whether we can use link information to improve classification accuracy. Our approach is to propagate category probabilities over linked pages. We find that using link information leads to marginal improvements over a baseline that uses a Naive Bayes model. For the initially misclassified pages, link information is either not available or contains too much noise.

1 Introduction

Previous years of the XML mining track have explored the utility of using XML document structure for classification accuracy. It proved to be difficult to obtain better performance [1]. This year the data consists of a collection of wikipedia XML documents that have to be categorized into fairly high-level wikipedia categories and the link structure between these documents. Link structure has been found to be a useful additional source of information for other tasks such as ad hoc retrieval [2] and entity ranking [4]. Our aim at the XML Mining Track is to examine whether link structure can also be exploited for this classification task.

2 Classification Model

For our baseline classification model we use a classical Naive Bayes model [3]. The probability of a category given a document is:

$$P(cat|d) = \frac{P(d|cat) * P(cat)}{P(d)} \tag{1}$$

Since $P(d)$ does not change over the range of categories we can omit it. For each document the categories are ranked by their probabilities, and the category with the highest probability is assigned to the document:

$$\begin{aligned} ass\_cat(d) &= \arg\max_{cat \in cats} P(d|cat) * P(cat) \\ &= \arg\max_{cat \in cats} P(t_1|cat) * P(t_2|cat) * .. * P(t_n|cat) * P(cat) \end{aligned} \tag{2}$$

where $t_i ... t_n$ are all terms in a document. The probability of a term occurring in a category is equal to its term frequency in the category divided by the total number of

terms in the category. Feature (term) selection is done according to document frequency. We keep 20% of the total number of features [5]. We propagate category information over links as follows:

$$\begin{aligned} P_0(cat|d) &= P(cat|d) \\ P_1(cat|d) &= \sum_{d \to d'} P(d|d')P(cat|d') \\ P_2(cat|d) &= \sum_{d' \to d''} P(d|d'')P(cat|d'') \end{aligned} \tag{3}$$

where d' consist of all documents that are linked to or from d, and d'' are all documents that are linked to or from all documents d'. The probabilities are uniformly distributed among the incoming and/or outgoing links. The final probability of a category given a document is now:

$$P'(cat|d) = \mu P_0(cat|d) + (1-\mu)(\alpha P_1(cat|d) + (1-\alpha)P_2(cat|d)) \tag{4}$$

The parameter μ determines the weight of the original classification versus the weight of the probabilities of the linked documents. Parameter α determines the weight of the first order links versus the weight of the second order links.

3 Experimental Results

Documents have to be categorized into one of fifteen categories. For our training experiments, we use 66% of the training data for training, and we test on the remaining 33%. Throughout this paper we use accuracy as our evaluation measure. Accuracy is defined as the percentage of documents that is correctly classified, which is equal to micro average recall. Our baseline Naive Bayes model achieves an accuracy of 67.59%. Macro average recall of the baseline run is considerably lower at 49.95%. All documents in the two smallest categories are misclassified. Balancing the training data can improve our macro average recall.

When we use the link information we try three variants: do not use category information of linked data, use category information of the training data, and always use category information of linked data. Other parameters are whether to use incoming or outgoing links, μ and α. For parameter μ we tried all values from 0 to 1 with steps of 0.1, only the best run is shown. The results are given in Table 1. The accuracy of the runs using link information is at best only marginally better than the accuracy of the baseline. This means that the difficult pages, which are misclassified in the baseline model, do not profit from the link information. The links to or from pages that do not clearly belong to a category and are misclassified in the baseline run, do not seem to contribute to classification performance. These linked pages might also be more likely to belong to a different category.

On the test data we made two runs, a baseline run that achieves an accuracy of 69.79%, and a run that uses in- and outlinks, $\alpha = 0.5$ and $\mu = 0.4$, with an accuracy of 69.81%. Again the improvement in accuracy when link information is used is only marginal.

Table 1: Training Classification results

		Inlinks		Outlinks		In- and Outlinks	
Link info	α	μ	Accuracy	μ	Accuracy	μ	Accuracy
Baseline			0.6759		0.6759		0.6759
None	0.75	1.0	0.6759	1.0	0.6759	1.0	0.6759
None	1.0	1.0	0.6759	1.0	0.6759	1.0	0.6759
Training	0.5	0.5	**0.6793**	0.4	0.6777	0.4	0.6819
Training	0.75	0.5	**0.6793**	0.5	0.6777	0.5	0.6806
Training	1.0	0.6	0.6780	0.5	0.6780	0.6	0.6777
All	0.5	0.5	0.6780	0.3	0.6816	0.4	**0.6858**
All	0.75	0.6	0.6780	0.3	0.6848	0.5	0.6819
All	1.0	0.6	0.6784	0.4	**0.6858**	0.6	0.6787

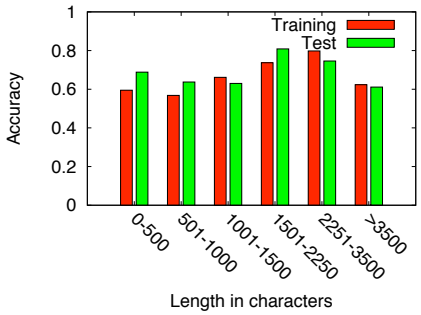

Fig. 1: Accuracy vs. document length

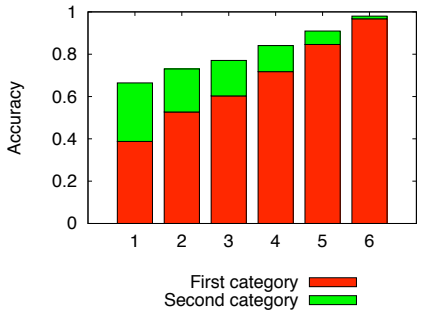

Fig. 2: Accuracy of first two categories

4 Analysis

We try to analyze on which kind of pages the most errors are made in our baseline run. Considering the length of pages, shorter pages do not tend to be more difficult than longer pages as can be seen in Fig. 1. When the output probabilities of the two highest scoring categories lie close together, the page is more likely to be misclassified. This is shown in Fig. 2 where we divided the training data over 6 bins of approximately the same size sorted by the fraction (P_{cat1}/P_{cat2}).

In our baseline run pages without links also seem to get misclassified more often than pages with in- and/or outlinks (see Fig. 3). When link information is available, and we try to use it, there are two sources of error. The first source of error, is that not all linked pages belong to the same category as the page to classify (see Table 2). However, when we classify pages that have links using only the link information, there are some cases where the accuracy on these pages is well above the accuracy of the complete set. To obtain our test data we have used both incoming and outgoing links, which means that almost half of the pages do not belong to the same category as the page to classify. Secondly, we only know the real categories of the pages in the training data, which is only 10% of all data. For all pages in the test data, we estimate the probability of

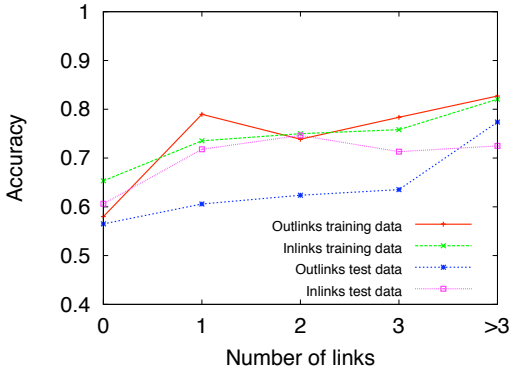

Fig. 3: Accuracy vs. number of links

Table 2: Statistics of training and test data

	# pages				links	% links with same cat.		
Data	total	with inlinks	with outlinks	with links	/page	inlinks	outlinks	links
Training	11,437	2,627 (23%)	5,288 (46%)	5,925 (52%)	0.7	76.8%	41.1%	45.8%
Test	113,366	88,174 (77%)	103,781 (91%)	107,742 (94%)	5.6	77.2%	53.4%	59.0%

each category belonging to that page. With a classification accuracy of almost 70%, this means we introduce a large additional source of error.

5 Conclusion

It is difficult to use link information to improve classification accuracy. A standard Naive Bayes model achieves an accuracy of almost 70%. While link information may provide supporting evidence for the pages that are easy to classify, for the difficult pages link information is either not available or contains too much noise.

Acknowledgments This research is funded by the Netherlands Organization for Scientific Research (NWO, grant # 612.066.513).

REFERENCES

[1] L. Denoyer and P. Gallinari. Report on the xml mining track at inex 2007 categorization and clustering of xml documents. *SIGIR Forum*, 42(1):22–28, 2008.

[2] J. Kamps and M. Koolen. The importance of link evidence in Wikipedia. In *Advances in Information Retrieval: ECIR 2008*, pages 270–282, 2008.

[3] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1–47, 2002.

[4] A.-M. Vercoustre, J. Pehcevski, and J. A. Thom. Using wikipedia categories and links in entity ranking. In *Focused Access to XML Documents*, pages 321–335, 2007.

[5] K. Williams. Ai::categorizer - automatic text categorization. Perl Module, 2003.