A probabilistic description-oriented approach for categorising Web documents*

Norbert Gövert Mounia Lalmas† Norbert Fuhr
University of Dortmund
{goevert,mounia,fuhr}@ls6.cs.uni-dortmund.de

Abstract

The automatic categorisation of web documents is becoming crucial for organising the huge amount of information available in the Internet. We are facing a new challenge due to the fact that web documents have a rich structure and are highly heterogeneous. Two ways to respond to this challenge are (1) using a representation of the content of web documents that captures these two characteristics and (2) using more effective classifiers.

Our categorisation approach is based on a probabilistic description-oriented representation of web documents, and a probabilistic interpretation of the k-nearest neighbour classifier. With the former, we provide an enhanced document representation that incorporates the structural and heterogeneous nature of web documents. With the latter, we provide a theoretical sound justification for the various parameters of the k-nearest neighbour classifier.

Experimental results show that (1) using an enhanced representation of web documents is crucial for an effective categorisation of web documents, and (2) a theoretical interpretation of the k-nearest neighbour classifier gives us improvement over the standard k-nearest neighbour classifier.

*This work has been carried out in the framework of the EuroSearch project, LE4-8303.

†Now at Department of Computer Science, Queen Mary & Westfield College, University of London.

1 Introduction

In May 1998, the publicly accessible part of the Internet contained 320 million pages. The figure is continually increasing. In this exploding scenario, subject-oriented browsing tools have become a vital method for accessing the huge amount of information available on the Internet. For example, the international *Yahoo!* (`http://www.yahoo.com/`) or the German *DINO-Online* (`http://www.dino-online.de/`) services partition the web information space into subject categories meaningful to web users. Web documents are then classified according to this scheme.

In most if not all subject-oriented browsing tools, the classification is done manually, partly by the maintainers of the services and partly by the users or information providers themselves. The main problems with manual classifications are that (1) they require a vast amount of intellectual work, and (2) they suffer from the fact that a significant part of the database content is either outdated or inconsistent. There is therefore the need to perform an *automatic categorisation* of web documents.

A number of algorithms to categorise text documents have been developed, and experimented with (see, for example, [Yang 99] for a survey and evaluation of text categorisation approaches). In these approaches, keywords are extracted from text documents, and documents are represented as vectors of weighted terms, where the weights are computed based on the widely used $tf \times idf$ indexing [Salton & Buckley 88]. These approaches performed well when applied to texts of a common domain and with an homogeneous structure (in standard information retrieval, documents are considered as atomic units).

Web documents have a richer structure (e. g., they are composed of separate units, they are linked to other documents, they provide markup for headings and highlighting of terms) and are highly heterogeneous (e. g., they contain units of varying discourses, their content

is not specific to a particular domain). Representing web documents using the standard $tf \times idf$ measure cannot capture this structural and heterogeneous nature. Therefore, the approaches used in information retrieval to categorise texts cannot be applied *as they are* to categorise web documents.

One possible approach is to enhance the representation of web documents to include the ***structural and heterogeneous nature*** of web documents. The enhanced representation would take into account features specific to web documents, as well as features standard to text documents. It can then be used as the basis for categorising web documents. This is the approach followed in our work.

We use a probabilistic regression method initially proposed in [Fuhr & Buckley 91] to obtain a representation that reflects the nature of web documents. This strategy is a *description-oriented* indexing approach; it takes into account (1) features specific to web documents (e. g., a term appears in a title, a term is highlighted, a document is linked to another document), as well as (2) features standard to text documents (e. g., term frequency). This strategy is a kind of long-term learning process that collects feedback from available data (a test-bed of pre-categorised documents is used for that purpose), thus aiming at an optimal representation of web documents for categorisation purposes.

On the basis of this enhanced representation of web documents, new documents can then be classified according to a pre-defined categorisation scheme. We propose a probabilistic interpretation of the k-nearest neighbour (kNN) classifier [Yang 94]. We chose kNN because it was shown in [Yang 99] to be very effective for the task of classifying single documents. In addition, kNN and its variants are computationally efficient. This is important since the web space is constantly increasing. With our probabilistic interpretation of kNN, we obtain a theoretical sound justification of the various kNN parameters. Our approach follows work of [Rijsbergen 89] and [Wong & Yao 95] who advanced that the idea of relevance in information retrieval can be viewed as a probabilistic inference process.

We have evaluated our approach using documents from a subset of the *Yahoo!* catalogue, namely the *Computers and Internet* category. Experimental results show that representations that incorporate features specific to the web lead to more effective categorisation of web documents. They also show that using the probabilistic interpretation of the kNN classifier gives us improvement over the standard kNN.

This paper is divided into the following sections. In Section 2, we describe how documents are represented using our probabilistic description-oriented approach. In Section 3, we present the classifier which is based on a probabilistic interpretation of kNN. In Section 4, we describe our experiments, together with the evaluation procedure. In Section 5, we present and discuss our results. We also compare our results to those obtained in related work. Finally, in Section 6, we conclude with some thoughts for future work.

2 Document representation

To allow for an effective categorisation of web documents, we use the description-oriented probabilistic indexing approach developed in [Fuhr & Buckley 91] but adapted to document categorisation.

As in standard information retrieval indexing, we represent documents as vectors of weighted terms. The difference between a standard indexing approach and our indexing approach is that in the former the term weights are computed based on the widely used $tf \times idf$, whereas in the latter, they are computed based on the so-called ***relevance description*** $\vec{x}(t,d)$ for term-document pair (t,d). $\vec{x}(t,d)$ is a vector composed of features $x_1(t,d), x_2(t,d), \ldots$ that are considered important for the task of assigning weights to terms with respect to web document categorisation. Examples of features are: $x_1(t,d) = 1$ if t is the most frequent term in d, otherwise $x_1(t,d) = 0$; $x_5(t,d) = 1$ if the term t appears in the title of the document d, otherwise $x_5(t,d) = 0$.

With a description-oriented approach, we can derive a term weighting scheme specific to web documents; the structural and heterogeneous nature of web documents can be captured (e. g., text data, lists of anchors, documents linked to other documents). Furthermore, the approach can be applied with little effort to web documents written in any language. Obviously, we would need stemmers and stop word lists specific to other languages, but other aspects of the indexing process are language-independent.

The term weights are computed by estimating the probability $P(R|\vec{x}(t,d))$. In [Fuhr & Buckley 91], relevance data is used to estimate $P(R|\vec{x}(t,d))$. For an indexing task, the relevance is based on query-document relationships (which document is relevant to which query), whereas for a categorisation task, the relevance data is based on category-document relationships (which document belongs to which category). Therefore the original interpretation of $P(R|\vec{x}(t,d))$ given in [Fuhr & Buckley 91] must be adapted to our categorisation task. For arbitrary pairs of documents d and d', we consider the sets of terms that occur in both documents. $P(R|\vec{x}(t,d))$ denotes the probability that a description of term t in document d occurs in another document d' belonging to the same category as d.

Following this view, we describe how the probability $P(R|\vec{x}(t,d))$ is computed (for more details, see [Fuhr & Buckley 91]). $P(R|\vec{x}(t,d))$ is derived from a learning sample (relevance data) $L \subset D \times D \times \mathcal{R}$ where:

- D is the set of pre-categorised documents (test-bed);
- $\mathcal{R} = \{R, \bar{R}\}$ for relevant and not relevant[1];
- $L = \{(d, d', r(d,d'))|d, d' \in D\}$ such that:

$$r(d,d') = \begin{cases} R & \text{if } d \text{ and } d' \text{ belong to the} \\ & \text{same category,} \\ \bar{R} & \text{otherwise.} \end{cases}$$

Based on L, we form a multi-set of relevance descriptions with relevance judgements

$$L^x = [(\vec{x}(t,d), r(d,d'))|t \in d \cap d' \wedge (d, d', r(d,d')) \in L]$$

This set with multiple occurrences of elements (bag) forms the basis for the estimation of $P(R|\vec{x}(t,d))$.

The values $P(R|\vec{x}(t,d))$ are estimated by applying probabilistic classification procedures as developed in pattern recognition and machine learning because they use additional (plausible) assumptions to compute the estimates. The classification procedure yielding an estimation of the probabilities $P(R|\vec{x}(t,d))$ is termed an *indexing function* $e(\vec{x}(t,d))$.

Let $y(d,d')$ denote a class variable representing the relevance judgement $r(d,d')$ for each element of L. Now we seek for a regression function $\vec{e}_{opt}(\vec{x})$ which yields an optimal approximation of the class variable y. As optimisation criterion, minimum squared errors are used (E denotes the expectation):

$$E((y - e_{opt}(\vec{x}))^2) \stackrel{!}{=} min$$

To derive an (optimal) indexing function from the learning sample L^x we use the ***least square polynomials*** approach [Knorz 83] [Fuhr 89], which was shown effective in [Fuhr & Buckley 93]. In this approach polynomials with a predefined structure are taken as function classes. Based on the relevance description in vector form $\vec{x}$, a polynomial structure $\vec{v}(\vec{x}) = (v_1, \ldots, v_L)$ has to be defined:

$$\vec{v}(\vec{x}) = (1, x_1, x_2, \ldots, x_N, x_1^2, x_1 x_2, \ldots)$$

Here N denotes the number of dimensions of $\vec{x}$. In practice, mostly linear and quadratic polynomials are regarded. The indexing function now yields:

$$e(\vec{x}) = \vec{a}^T \cdot \vec{v}(\vec{x})$$

where $\vec{a} = (a_i)$ for $i = 1, \ldots, L$ is the coefficient vector to be estimated. For example, if we take the linear polynomial structure $\vec{v}(\vec{x}) = (1, x_1, x_2)$, then $\vec{a} = (a_1, a_2, a_3)$ and the indexing function is $e(\vec{x}(t,d)) = a_1 + a_2 x_1 + a_3 x_2$.

The coefficient vector $\vec{a}$ is computed by solving the following linear equation system [Schürmann 77]:

$$E(\vec{v} \cdot \vec{v}^T) \cdot \vec{a} = E(\vec{v} \cdot y)$$

As an approximation for the expectations, the corresponding arithmetic means from the learning sample are taken.

Once the parameter vector $\vec{a}$ is derived from the training set, it can be used for indexing both the documents in the test-bed (pre-categorised documents) and those to be classified.

The output of the procedure consists of a database of triplets (t, d, w) of term t, document d, and weight w, where $w = P(R|\vec{x}(t,d))$ is the weight of term t in the document.

3 Document classifier

Using the probabilistic description-oriented representation of documents, new documents can be assigned categories. This task is performed using kNN or *k-nearest neighbour classifier* which works as follows [Yang 94]. Given an arbitrary document, the method ranks its nearest neighbours among training documents (a test-bed of pre-categorised documents), and uses the categories of the k top-ranked documents to predict the category of the new document. The similarity score of each (k) neighbour document is used as a weight of its categories, and the sum of category weights over the k nearest neighbours are used for category ranking.

We use a probabilistic interpretation of kNN; we compute the probability that a document d belongs to category c. This probability is viewed as the probability that d implies c, which is estimated as follows:

$$P(d \rightarrow c) \approx \sum_{d' \in NN} P(d'|NN) \cdot P(d \rightarrow d') \cdot P(d' \rightarrow c)$$

where NN is the set of nearest neighbour documents, $P(d'|NN)$ is the normalisation factor[2] and $P(d \rightarrow d')$ corresponds to the similarity between d and d'. The factor $P(d' \rightarrow c)$ reflects the categorisation information available for d' (which was used to build the indexing function, as described in Section 2):

$$P(d' \rightarrow c) = \begin{cases} 1 & \text{if } d' \text{ belongs to } c, \\ 0 & \text{otherwise.} \end{cases}$$

[1] The method can be generalised to include a wider relevance scale.

[2] $\sum_{d' \in NN} P(d'|NN) = 1$

Following [Wong & Yao 95] $P(d \rightarrow d')$ is computed as:

$$\begin{aligned} P(d \rightarrow d') &= \sum_t P(d \rightarrow t)P(t \rightarrow d') \\ &= \sum_t P(t|d)P(d'|t) \\ &= \sum_t \frac{P(d|t)P(t)}{P(d)} P(d'|t) \end{aligned}$$

where $P(t)$ reflects the probability of a term, and is approximated by the inverse document frequency (idf) of the term. $P(d)$ corresponds to a normalisation factor with respect to the document to be categorised (i. e., $\sum_t P(t|d) = 1$). The probability $P(d|t)$ ($P(d'|t)$) reflects the indexing weights of term t in document d (d', respectively).

We can see that with our probabilistic interpretation of the kNN classifier, the role of a term differs whether it occurs in a document of the test-bed or a document to be classified. In the former case, the role is expressed with a probability $P(d \rightarrow t)$, whereas in the latter, it is expressed with a probability $P(t \rightarrow d')$. As opposed to this the standard kNN uses a *symmetric* measure for the degree of similarity of a document d to another document d', i. e., a term in the document to be classified is treated in the same way as in a document of the test-bed.

This interpretation of kNN can be compared to the notion of general imaging [Rijsbergen 89]. With general imaging, the probability that d implies c, $P(d \rightarrow c)$, is computed by means of the probability $P(d' \rightarrow c)$ for those documents d' that are closest (the imaging process) to d as given by $P(d \rightarrow d')$.

4 Experiments

The categorisation method described in this paper has been evaluated using the documents present under the *Computers and Internet* category of the *Yahoo!* catalogue, and all its direct and indirect sub-categories. Our approach requires a test-bed of pre-categorised documents (for the learning phase). The creation of the test-bed is described in Section 4.1. The method used to evaluate our approach is described in Section 4.2. The baseline with which we compare our approach is given in Section 4.3. The settings of our experiments are described in Section 4.4.

4.1 Creation of the test-bed

Documents from *Yahoo!*'s *Computers and Internet* catalogue were used. Documents directly referenced by *Yahoo!* categories (referred to as root documents) are often a list of entry points to a group of documents with very little content[3], so the documents referenced by the documents directly referenced by *Yahoo!* categories were also used.

Table 1 shows various statistics[4] on the test-bed collection. A leaf document is a document found under a leaf category of *Yahoo!*'s *Computers and Internet* catalogue, whereas an inner document is one found under an inner (non-leaf) category. We have a total of 2806 categories and 18639 documents, which means that we have a mean of approximately 6.6 documents per category. This number may be changed by merging categories together if such a number does not allow for effective learning of our categorisation approach. This is possible because of the hierarchical nature of the *Yahoo!* catalogue.

Only textual data was considered for indexing purpose. Therefore, images, speech, Java scripts and applets, forms, etc. were removed. A document directly referenced by a *Yahoo!* category (*root document*) was indexed on the basis of that document and the documents it links to. This was done to ensure high recall. Finally, to ensure high precision, only referenced documents that are on the same web site as the document referred by *Yahoo!* categories were considered. We call the root document together with documents on the same server directly referenced by the root document the *radius1 document*.

4.2 Evaluation method

Evaluating our categorisation tool consists of determining whether the appropriate categories are assigned to new documents. To determine this, we split the test-bed into a learning sample L and a test set T (the two sets are disjoint). The documents composing the test set are treated as the new documents to classify, whereas those composing the learning set are used for developing the indexing function, as described in Section 2. Since the documents in the test set also have categories, it can be verified whether the categories induced by our approach are the same as those originally assigned to the documents.

We define $(ca)_{ij}$ where ca_{ij} is 1 if category c_i has been assigned to document d_j (of the test-bed), and 0 otherwise. The aim is to determine whether kNN assigns or not a category c_i to a document d_j of the test set, this assignment being denoted a_{ij}, such that $a_{ij} = ca_{ij}$.

[3] This is especially true for documents that only contain a frameset with pointers to documents that fill the frames.

[4] *Yahoo!* changes over time. Our test-bed has been frozen on June, 1998.

	categories			documents			documents/category		
level	inner	leaf	total	inner	leaf	total	inner	leaf	total
0	1	0	1	0	0	0	0.0	-	0.0
1	22	3	25	472	20	492	21.5	6.7	19.7
2	123	171	294	1817	964	2781	14.8	5.6	9.5
3	196	316	512	3084	1804	4888	15.7	5.7	9.5
4	216	529	745	2857	2660	5517	13.2	5.0	7.4
5	157	488	645	1463	1741	3204	9.3	3.6	5.0
6	73	330	403	213	1083	1296	2.9	3.3	3.2
7	21	118	139	57	287	344	2.7	2.4	2.5
8	0	42	42	0	117	117	-	2.8	2.8
total	809	1997	2806	9963	8676	18639	12.3	4.3	6.6

Table 1: Test-bed statistics

The evaluation is based on the classic notions of precision and recall but adapted to text categorisation: *precision* is the probability that, if d_j is categorised under c_i ($a_{ij} = 1$), this decision is correct ($ca_{ij} = 1$); *recall* is the probability that, if d_j should be categorised under c_i ($ca_{ij} = 1$), this decision is taken ($a_{ij} = 1$).

We perform a micro-averaging evaluation: the effectiveness is obtained by globally summing over all individual decisions. Let C be the set of all categories. The precision and recall values are therefore given by:

$$Recall = \frac{\sum_{c_i \in C, d \in T} a_{ij} \cdot ca_{ij}}{\sum_{c_i \in C, d \in T} ca_{ij}}$$

$$Precision = \frac{\sum_{c_i \in C, d \in T} a_{ij} \cdot ca_{ij}}{\sum_{c_i \in C, d \in T} a_{ij}}$$

To be able to consider not only the percentage of correct decisions, but also how categories are ranked according to kNN, first we merge all the ranked categories out of kNN of all documents (this is possible because the weights leading to the ranking are normalised). Then, we compute the precision and recall values as done in information retrieval; here the relevance assessments correspond to the correct decisions.

In addition, to compare our results to others, we apply the same procedure, but we consider only the first ranked categories. By doing this, we reflect the decision as made by kNN.

4.3 Baseline

We require two baselines, one with which to compare the effectiveness of our description-oriented document representation for categorisation purposes, and a second one with which to compare the effectiveness of our probabilistic interpretation of the kNN classifier.

For the first baseline, documents are presented by vectors of weighted terms, where the weights are given by a standard $tf \times idf$ indexing [Salton & Buckley 88].

For the second baseline, we use the following standard formulation of kNN ([Yang 94]). The similarity between a document d to be categorised and a category c is computed as

$$sim'(d, c) = \sum_{d' \in NN} sim(d, d') \cdot c(d')$$

where NN is the set of the k documents d' for which $sim(d, d')$ (the similarity between d and d') is maximum. The function $c(d')$ yields 1 if document d' belongs to category c, and 0 otherwise. The document d is categorised by the category c with the highest $sim'(d, c)$ value. $sim(d, d')$, the similarity score between documents, is the cosine function [Salton & Buckley 88].

4.4 Settings for the experiments

We describe the settings of our experiments. Each parameter is given along with its domain, i. e. its set of possible values.

Document { `root`, `radius1` }. The representation of a document can be based on the web page directly referenced by *Yahoo!* only (`root` strategy), or on the root document and those it links to on the same web site (`radius1` strategy).

Category { `top`, `all` }. The outcome of any experiments can be evaluated in several ways. The evaluation can be based on perfect match (`all`) or partial match. An extreme strategy in the partial match case is to consider the top 25 categories only (`top`). In our experiments we consider the top 25 categories only, since the statistics showed that the number of documents per category in our test-bed is very low.

Relevance description vector We have defined ten attributes forming the relevance description vector:

- $x_1(t,d) = 1$ if term t is the most frequent term in document d; otherwise $x_1(t,d) = 0$;
- $x_2(t,d)$ is the number of terms in document d;
- $x_3(t,d)$ is the number of distinct terms in document d;
- $x_4(t,d)$ is the frequency of term t in document d;
- $x_5(t,d) = 1$ if term t appears in the title of document d; otherwise $x_5(t,d) = 0$;
- $x_6(t,d) = 1$ if term t is highlighted in document d (bold, emphasised, etc.); otherwise $x_6(t,d) = 0$;
- $x_7(t,d) = 1$ if term t appears in a heading in document d; otherwise $x_7(t,d) = 0$;
- $x_8(t,d) = 1$ if term t appears in the first paragraph of document d; otherwise $x_8(t,d) = 0$;
- $x_9(t,d) = 1$ if t appears in the root node of document d; otherwise $x_9(t,d) = 0$;
- $x_{10}(t,d)$ is the inverse document frequency for term t, that is the number of documents in which t occurs.

Note that some features depend on both the term and the document, whereas others depend only on the term, or on the document. Some features may be better than others for deriving the probabilistic indexing weights for web documents. We can use any subset of the ten features. Also, the computation of a feature can be modified, for example to include normalisation and logarithm values, which are known to increase indexing effectiveness in information retrieval.

Polynomial structure { *power set of set of components of complete quadratic polynomial* }. The set of components of the complete quadratic polynomial depends on the dimension of the relevance description vector. In our experiments, we use a linear polynomial structure.

Query term selection For efficiency reasons, we do not consider all terms in large query documents when categorising such documents. Therefore only the top 50 terms are taken (i.e. those terms with the highest query term weights).

5 Results and analysis

To obtain some initial results about the effectiveness of our categorisation approach, we carried out a number of experiments. The results are presented first. We analyse them in Section 5.4.

We considered about 70 % of the test-bed as learning documents (used in the learning phase); the other 30 % of the documents were taken as test documents, i.e. as input to the kNN classifier.

We decided to set the category parameter for our preliminary evaluation to `top` because setting them to `all` leads to very low effectiveness for both the baseline indexing and our probabilistic indexing. The average precision for the baseline is 2.45 % and for the probabilistic description-oriented indexing 2.13 %. With such low results, proper comparison, and hence enhancement, is not possible. Only results that are either positive or allow us to draw some conclusions are presented and discussed.

5.1 Probabilistic indexing vs. baseline indexing

Figure 1 shows the results obtained with our description-oriented indexing approach and the baseline $tf \times idf$ indexing. Here we considered the `radius1` documents. The average precision for the probabilistic indexing is 31.39 % and for the baseline 31.05 %. For the first ranked categories we obtain a precision of 36.5 % for the probabilistic indexing and for the baseline 33.57 %.

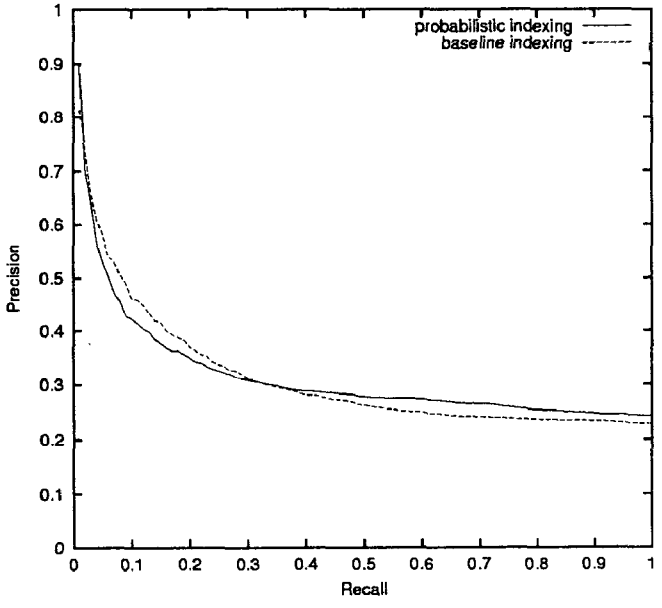

Figure 1: Probabilistic indexing vs. baseline indexing, `radius1`

5.2 Probabilistic vs. standard kNN

In Figure 2 we compare results obtained with our probabilistic interpretation of kNN (Section 3) to results ob-

tained with the standard kNN (Section 4.3). The average precision for our interpretation of kNN is 26.81 %, whereas it is 25.94 %. For the first ranked categories we obtain a precision of 15.45 % for the probabilistic kNN and 13.18 % for the standard kNN. This experiment has been performed considering the `root` document nodes only.

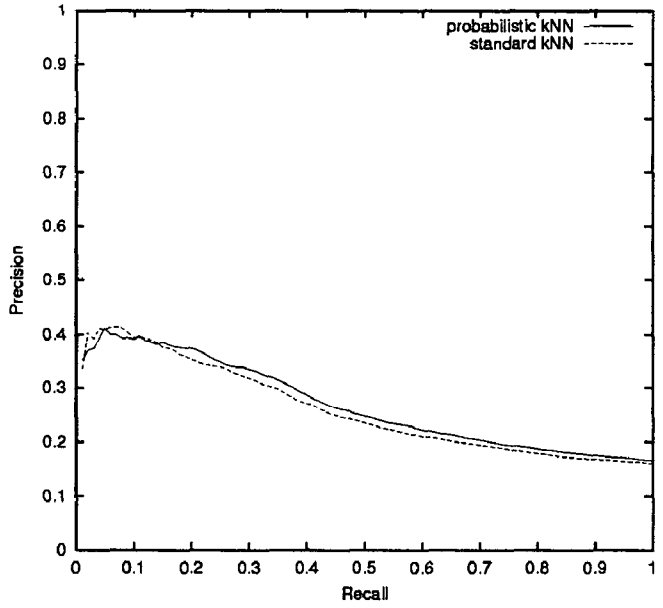

Figure 2: Probabilistic vs. standard kNN, `root`

5.3 `radius1` vs. `root` documents

In Figure 3 we compare the results obtained from using `radius1` documents to results obtained from using `root` documents. For both experiments we use our probabilistic indexing as well as the probabilistic interpretation of kNN. For `radius1` we reach an average precision of 31.39 % (first-rank precision 36.5 %), for `root` a precision of 26.81 % is achieved (first-rank precision 15.45 %).

5.4 Analysis an comparison to related work

Our results are poor when compared to those obtained in text categorisation applied to standard information retrieval test collections. For example with the *Reuters text categorisation test collection*[5] people achieve average precision values between 70 and 80 % (see e. g. [Yang 99]). However recent work from [Chakrabarti et al. 98] shows a precision of 32 % for a term-based classifier applied to a part of the *Yahoo!* catalogue. We can see that specific problems occur for the automatic categorisation of web documents:

[5] `http://www.research.att.com/~lewis/reuters21578.html`

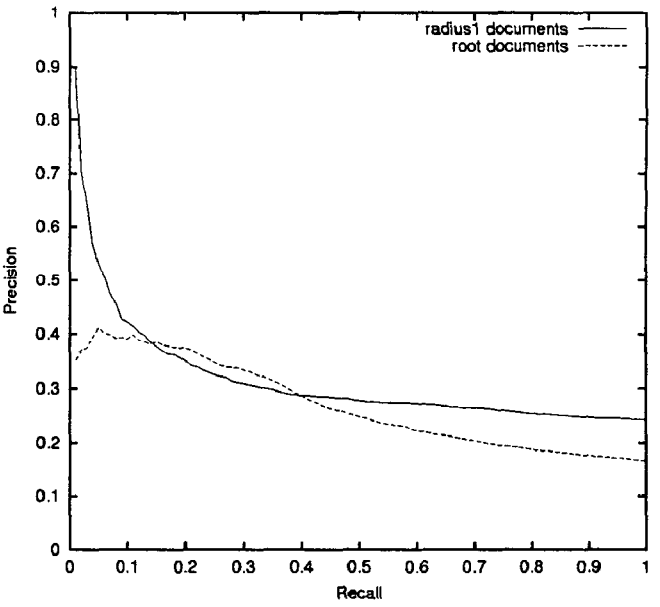

Figure 3: `radius1` vs. `root` documents

Low quality of web documents The quality of web documents is low compared to that of documents used in a controlled environment (e. g. papers in a proceedings). Web documents contain more typing errors and can be written in a very sloppy style.

Inconsistency of classification The classification of the documents in *Yahoo!* (our test-bed) is not done as rigorously as that in other subject-oriented browsing tools (e. g., news from the Reuters news agency).

We can still draw some specific conclusions regarding our approach for automatically categorising web documents.

In earlier experiments (not presented here), we categorised web documents with respect to 2806 categories. The results were very low (very few documents were correctly categorised). The average precision of the baseline indexing and the probabilistic indexing was low; in addition the description-oriented approach did not perform significantly better than the baseline method. The main reason for this is that the learning sample is small (it consists of 11,699 documents) which means that the indexing function has to be developed from approximately 4.2 documents per category.

This means that the learning sample (the set L) does not provide enough indicative relevance data to build the indexing function. Most pairs of documents d and d' that have common terms belong to different categories, so the values $r(d, d')$ are mostly set to $\bar{R}$. We have a very small number of pairs of documents d and d' such that $r(d, d') = R$. To allow for a better train-

ing phase, we need to adopt a merging strategy. We are currently investigating other ways to construct the learning sample that contains more instances of pairs of documents d and d' where $r(d, d') = R$. One approach that is currently being implemented is to set $r(d, d')$ to R if (1) d and d' belong to the same category and (2) the category associated with d is a direct or indirect sub-category of d' and vice-versa.

When we evaluate our results using the `top` categories we can see that our experiments show promising results. We can already see that using our probabilistic indexing as a basis to categorise web documents gives slightly better results than the standard $tf \times idf$ indexing. Not only we have a theoretical justification for our probabilistic retrieval function, but also experimental indication that this retrieval function is effective. Furthermore, the results obtained with the `radius1` and `root` node indexing show that, for categorisation purposes, a web document should be indexed by considering its content and the content of web documents that are linked from it.

6 Conclusion and future work

In this paper, we presented and evaluated an approach to automatically categorise web documents. Our approach is based on a probabilistic description-oriented indexing: it takes into account features specific to web documents, and features standard to text documents. Using the probabilistic indexing, new documents are categorised using a probabilistic interpretation of the k-nearest neighbour classifier.

Our main conclusion is that our probabilistic description-oriented indexing approach promises effective results, although further experiments are needed to refine our categorisation tool. Many variants of our approach can be experimented with. In particular, (1) choice of the polynomial structure, (2) selection and calculation of the features, and (3) deriving more indicative data for developing the indexing function. Note that these are only possible with our description-oriented approach, thus giving us more scope to refine our categorisation tool.

References

Chakrabarti, S.; Dom, B.; Indyk, P. (1998). Enhanced Hypertext Categorization Using Hyperlinks. In: Haas, L.; Tiwary, A. (eds.) : *Proceedings of the 1998 ACM SIGMOD. International Conference on Management of Data.* ACM Special Interest Group on Management of Data, ACM, New York.

Fuhr, N.; Buckley, C. (1991). A Probabilistic Learning Approach for Document Indexing. *ACM Transactions on Information Systems 9(3)*, pages 223–248.

Fuhr, N.; Buckley, C. (1993). Optimizing Document Indexing and Search Term Weighting Based on Probabilistic Models. In: Harman, D. (ed.) : *The First Text REtrieval Conference (TREC-1)*, pages 89–100. National Institute of Standards and Technology Special Publication 500-207, Gaithersburg, Md. 20899.

Fuhr, N. (1989). Models for Retrieval with Probabilistic Indexing. *Information Processing and Management 25(1)*, pages 55–72.

Knorz, G. (1983). *Automatisches Indexieren als Erkennen abstrakter Objekte.* Niemeyer, Tübingen.

van Rijsbergen, C. J. (1989). Towards an Information Logic. In: Belkin, N.; van Rijsbergen, C. J. (eds.) : *Proceedings of the Twelfth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 77–86. ACM, New York.

Salton, G.; Buckley, C. (1988). Term Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management 24(5)*, pages 513–523.

Schürmann, J. (1977). *Polynomklassifikatoren für die Zeichenerkennung. Ansatz, Adaption, Anwendung.* Oldenbourg, München, Wien.

Wong, S.; Yao, Y. (1995). On Modeling Information Retrieval with Probabilistic Inference. *ACM Transactions on Information Systems 13(1)*, pages 38–68.

Yang, Y. (1994). Expert Network: Effective and Efficent Learning from Human Decisions in Text Categorisation and Retrieval. In: Croft, W. B.; van Rijsbergen, C. J. (eds.) : *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 13–22. Springer-Verlag, London, et al.

Yang, Y. (1999). An Evaluation of Statistical Approaches to Text Categorization. *Information Retrieval 1(1)*, pages 69–90.