Document Classification for Focused Topics

Russell Power, Jay Chen, Trishank Karthik, Lakshminarayanan Subramanian
(power, jchen, trishank, lakshmi)@cs.nyu.edu

Abstract

Feature extraction is one of the fundamental challenges in improving the accuracy of document classification. While there has been a large body of research literature on document classification, most existing approaches either do not have a high classification accuracy or require massive training sets.

In this paper, we propose a simple feature extraction algorithm that can achieve high document classification accuracy in the context of development-centric topics. Our feature extraction algorithm exploits two distinct aspects in development-centric topics: (a) most of these topics tend to be very focused (unlike semantically hard classification topics such as chemistry or banks); (b) due to local language and cultural underpinnings in these topics, the authentic pages tend to use several region specific features. Our algorithm uses a combination of *popularity* and *rarity* as two separate metrics to extract features that describe a topic. Given a topic, our output feature set comprises of: (i) a list of popular keywords closely related to the topic; (ii) a list of rare keywords closely related to the topic. We show that a simple joint classifier based on these two feature sets can achieve high classification accuracy while each feature sub-set in itself is insufficient. We have tested our algorithm across a wide range of development-centric topics.

1. Introduction

Document classification is a fundamental learning problem that is at the heart of many information management and retrieval tasks. In the context of development, document classification plays an important role for several application especially for organizing, classifying, searching and concisely representing large volumes of information. Given the high price of network connectivity, several research efforts [Pentland, Fletcher, and Hasson2004, Seth et al.2006, Jain, Fall, and Patra2004] have investigated new models for information access and distribution in developing regions by physically transporting hard disks or USBs or SD cards with large volumes of pre-loaded information relevant for the local region. For example, there has been a recent drive to establish information guides and portals for specific topics in the domains of agriculture, healthcare and education to improve operational practices on the ground in developing regions. The Blue Trunk Libraries project by WHO, Commcare, GuideViews and medical education modules are examples of specific information guides tailored to improve the training and education of healthcare workers in rural areas in developing regions.

This paper deals with the problem of document classification in relation to an ongoing project effort on *contextual information portals* that aims to build an information portal for any arbitrary topic based on information available on the Web. Given the wealth of information available online, we aim to construct a vertical slice of all related pages on the Web for a given topic.

Document classification is a critical component in building contextual information portals. Given an arbitrary topic, the goal is to determine all pages on the Web that is related to that topic. The focus of this paper is not on how to establish such a portal but on the specific sub-problem of classifying web pages for development-centric topics.

Document classification is an age-old problem in information retrieval which has been well studied. In the context of the Web, there is a large body of research literature on classification of web page content [Qi and Davison2009] using a variety of different approaches that leverage different information source types from the page: text, links, URL, hypertext and tags. Despite all these works, web page classification is still not a solved problem since existing approaches do not provide high levels of accuracy or require extensive training.

There are two main factors which make document classification a challenging problem: (a) feature extraction; (b) topic ambiguity. First, in any document classification algorithm, extracting the right set of features plays a critical role in determining the accuracy of classification. In text based classification, using standard textual similarity measures to compare and classify documents can often yield poor accuracy; Second, many broad topics are often ambiguous making classification of documents for these topics a hard problem. For ambiguous or broad topics the topic may have different meanings and the topic and its related terms may reappear in various contexts.

In this paper, we propose a simple feature extraction algorithm for development centric topics which when coupled with standard classifiers yields high classification accuracy. Our feature extraction algorithm exploits two distinct aspects in development-centric topics: (a) most of these topics tend to be very focused (b) due to local language and cultural underpinnings in these topics, the authentic pages tend to use several region specific features. The key takeaway message from this work is that due to the nature of development-centric topics, document classification becomes an easy problem if we extract the right feature set for each topic.

Our feature extraction algorithm uses a combination to two completely different and potentially opposing metrics to extract textual features for a given topic: (a) *popularity*; (b) *rarity*. Popularity of words related to a given word is commonly used across existing classifiers [Qi and Davison2009] to weight closely related terms within a document. Given a training set of documents related to the topic, the popularity metric determines a list of popular terms that are closely related to the topic.

Rarity is a metric that is particularly tailored for development-centric topics due to occurrence of region-specific or topic-specific rare terms across different documents. Given that most development-centric topics are focused topics with topic-specific underpinnings, the rarity metric can capture the list of rare terms that are closely related to the topic. To measure rarity of any given term (which need not be a single word but an n-gram), we leverage the Linguistic Data Consortium (LDC) data set to learn the frequency of occurrence of any n-gram on the Web. Though the LDC data set that we use is slightly old, we have found in a separate study that the relative rarity across most terms has been preserved over the past few years.

To achieve good classification accuracy, we need both these metrics for feature extraction. Either one by themselves may not provide good accuracy as has been attempted in prior classification studies based on popular keywords alone. In addition, restricting the feature set to only a combination of terms extracted using these two extreme metrics reduces the typical noise generated by using the entire text of a document for classification. For example, if a web page contains a large volume of text, the possibility of such pages being wrongly classified can be high. An additional advantage of our approach is the feature extraction process is very fast and simple and does not require extensive training sets.

We have tested our algorithm across a wide range of development-centric topics. Our results show that by combining our feature extraction algorithm with standard classifiers can results in very high accuracy of roughly 95% for recall of related documents and 99.95% precision in rejecting random documents unrelated to the topic.

2. Why not Text Similarity?

One approach to document classification that seems reasonable at first glance is to estimate the class of a document by clustering it based on a textual similarity metric. Cosine similarity is one frequently used metric for this purpose. For this task, we found a naive unweighted distance classification to yield very poor results. While this is somewhat to be expected, it highlights the importance of proper feature selection and weighting.

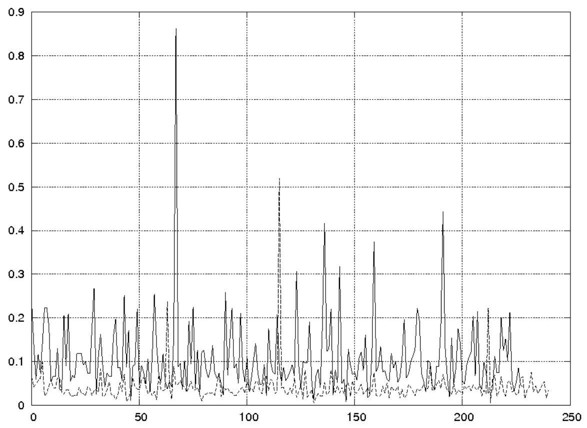

Figure 1: Cosine Similarity of Random Pages with Malaria Pages

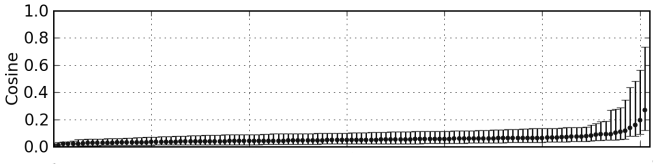

Figure 2: Cosine Similarity Distribution of Random Pages

As an example, Figures 1 shows the cosine similarity of random documents against those related to 'malaria'. Note that the correlation of malaria documents to random documents frequently exceeds the correlation within the topic group - there is no clear separation line between the two.

Figure 2 shows the cosine similarity within pages returned by a search for malaria. Though most pages share the expected small amount of similarity with one another, the noisiness of the signal makes accurate selection by this metric difficult.

The net result of this effect is that an arbitrary document can exhibit a high degree of correlation with a desired classification topic. This limits the effectiveness of any technique that uses document cosine similarity without additional information.

3. Related Work

A large amount of prior work has focused on document classification, and in particular web page classification.

3.1 Classifiers

The two techniques appearing most often for web page classification are naive Bayes and SVM learners. In recent years, other techniques have been proposed and used, with some success. [Nigam, Lafferty, and McCallum1999]

Naive Bayes approaches the problem of classification via a deceptively simplistic model: assume all features are independent of one another, and compute the class of a document based on maximal probability. The independence assumption allows for this to be evaluated easily by taking the simultaneous product of the conditional probabilities for each class. [Lewis1998]

Despite the simplicity of the approach, naive Bayes learners can routinely achieve results on par with more complex approaches. Research into this behavior has yielded the conclusion that, though the independence assumption is obviously untrue in many contexts, the resulting (possibly large) errors in probability do not prevent correct classification in many cases. [Domingos and Pazzani1996] [Zhang2004]

SVMs (Support Vector Machines) have been widely used for text classification in recent history [Joachims, Nedellec, and Rouveirol1998]. With the appropriate choice of kernel function, they can learn complicated separation functions, and can successfully operate on large datasets with appropriate optimizations. [Joachims1998]

3.2 Feature Extraction

Prior work for web page classification largely focuses on feature extraction and selection. In addition to the page text, additional components of pages and their relationships have been integrated to improve classification. We list a few of the major patterns here.

A common source of additional information is the HTML tags from the page; previous work has used titles and other tag data [Yang, Slattery, and Ghani2002] to label the text features. Other research involves using the geometry of the rendered HTML page to build up tree models based on the incoming link structure. [Shih and Karger2004].

A broad class of features for documents comes from anchors. Anchor-text from links pointing into a page can be associated with the page itself [Chakrabarti, Dom, and Indyk1998]. In addition, URLs and text from other pages may be accumulated into the candidate feature set. Some attempts to assign additional feature information eschew the traditional association of relatedness via anchors, and instead attempt to group together pages on the basis of their relative positions in a site tree or web graph (sibling-relationships). Still other approaches view the labeling of pages as a optimization problem on the graph of pages formed by anchors, and attempt to optimize labeling on the graph. [Angelova and Weikum2006]. [Qi and Davison2009] provides an overview of techniques that use the graph information to enhance classification.

Other work in the area has been on URL-only classifiers; these ask the question of whether is it feasible to classify a page knowing only the URL (and possibly some link structure) [Kan and Thi2005]. This is of particular interest for systems like web-crawlers and focused crawlers, where there is a desire to classify a page as desirable *before* retrieving it. Even with this restriction on available information, classification precision above 60% has been achieved on the WebKB dataset (described below). [Baykan et al.2009].

3.3 Testing

Testing is typically done using hand labeled data sets. One dataset commonly used for this purpose is the “4 University” set from WebKB [Craven et al.1998]. This consists of pages crawled from several universities, and grouped into seven categories: student, faculty, staff, course, project, department and other.

Due to the small size and ambiguity of some of the categories in this set, classification often is performed on a subset of the documents consisting of the student, faculty, staff and course groups. We follow this testing methodology here. Precision results of above 90% have been achieved on this reduced set. [Nigam, Lafferty, and McCallum1999].

4. Feature Extraction Algorithm

In this section, we describe our feature extraction algorithm for focused topics and how it can be used in conjunction with existing classifiers.

From our perspective, a *focused topic* by definition is unambiguous. In other words, given a topic, a human should be able to unambiguously state whether any document is related to that topic or not. However, defining ambiguity of a topic either mathematically or semantically is hard. For this purpose, we outline some of the typical properties satisfied by focused topics. First, the frequency of occurrence of the topic on the Web is not high. For example, fairly general topics such as “news”, “entertainment”, “media” appear in over 500 million pages and even topics such as “healthcare”, “chemistry” and “banking” appear in over 100 million pages on the Web. However, focused topics relevant in development contexts such as “rajinikanth”, “jackie chan”, “tuberculosis”, “malaria” “organic farming” appear in much fewer pages. Frequency alone is not sufficient to categorize a focused topic; one can have popular yet focused topics (such as “hiv”) or relatively rare terms which are ambiguous topics. Second, a focused topic does not have multiple distinctly different meanings in the dictionary. Third, and most importantly, given a set of topics of interest to a community, a topic is said to be focused within the list if the document overlap across topics is negligible if not null. For example, the possible document overlap between “malaria” and “Barack Obama” is small, while the document overlap between “baseball” and “mlb” is large.

Given a focused topic that satisfies these properties, the objective of our feature extraction algorithm is: *Given a focused topic and a training set of candidate authoritative pages on the topic, extract an appropriate feature set of textual terms that can be used in conjunction with any standard classifier to determine if any document is related to the topic or not.* Our feature extraction algorithm helps in condensing any document as a vector across the extracted feature set which in turn can be used by any classification algorithm such as Bayes or SVM.

Our extraction algorithm is also designed for *topic-specific* classifiers to determine documents corresponding to a single topic. While the algorithm can be extended for multi-topic classifiers (distinguishing across different topics), the feature extraction will not be well-suited for the case where the set of topics may be partially overlapping such as distinguishing between “malaria” and “cholera”.

Our algorithm uses two different and contrasting metrics to extract features from a text document: *popularity* and *rarity*. Given a training set of documents, we initially do some pre-processing and filtering of documents: we remove all documents that contain very little information and remove very popular terms in the document since they add significant noise to the classification. If the candidate set is generated by taking the top N pages from a search engine such as Google, we found that a non-trivial fraction of the top N documents contained very little text to aid in training the classifier.

To detect the relative frequency of a term t, we used the Linguistic Data Consortium dataset, which provides the web frequency of $n-$grams for $n \leq 5$. We denote this as $LDC(t)$. We use the LDC dataset to discount very popular terms (such as “the”, “and” etc.) from the feature set. A term in our description need not be a $1-$ gram but can be any $n-$ gram for $n \leq 5$.

For every term t, we compute the TF-IDF [Jones1972] value of that term as:

$$tfidf(t) = tf(t) \times log(N/N(t))$$

Here, $tf(t)$ represents the mean term frequency of t and $log(N/N(t))$ represents the inverse document frequency of term t where N is the overall number of documents and $N(t)$ is the number of documents t appears in. While the typical means to compute IDF is to use the candidate set, we use a more accurate estimate of IDF based on the LDC dataset. One obvious problem with using the candidate set for measuring IDF is that we anticipate the training set to be small and yield inaccurate values; for example, several topic-specific terms may appear in all documents with a measured IDF of 0. We use $LDC(t)$ as the estimate for $N(t)$ and the maximum of $N(t)$ across all terms in the LDC as our estimate of N.

We define a term to be a *popular term* related to the topic if the following constraint is met:

$$tfidf(t) > T_{th}, LDC(t) < P_{max}$$

Here, T_{th} is a lower bound on the TF-IDF value for a term to be considered. P_{max} is the upper bound on the LDC count to remove extremely popular terms from consideration. Based on manual inspection across different topics, we set $P_{max} = 100,000,000$ for the LDC dataset. Note that the LDC dataset that is currently publicly available is circa 2004 and may not be reflective of the currently Web frequency. However, for most terms, we have found the relative frequency to roughly remain similar over time: if t_1 was more popular than t_2 in 2004, that trend has continued for most terms. The value of T_{th} was also computed as a common base value across different topics. We compute the list of popular terms for different focused topics and have human experts determine appropriate thresholds for different topics. Across $15-20$ focused topics spanning different areas, we found $T_{th} = 4$ to be a good separation point across topics.

We use a graded measure to compute *rare terms*. The basic definition of a *rare term* is based on the following constraint:

$$tfidf(t) > R_{th}, LDC(t) < R_{max}$$

Here R_{th} is a lower bound on the tfidf value and is typically much smaller than T_{th} for popular terms. R_{max} is the upper bound on the LDC count to restrict this set to only consider rare terms. We need a lower bound on the tfidf based on R_{th} to remove all the rare terms which appear in very few documents in the candidate set and get picked up as related to the topic.

However, this basic definition of rarity is not sufficient. The basic problem is with how to set R_{max}. If we set R_{max} to a very low value such as 1000, then very few terms get selected and other important rare terms related to the topic are ignored. If we set R_{max} to a high value say $1,000,000$, then several terms not related to the topic get selected since R_{th} is very small and does not filter these terms.

Hence, we modified our rarity metric based on a graded measure. We defined a base low value of $R_{max} = 1000$ as the smallest value under consideration and divided the LDC scale based on a logarithmic scale; in other words, we considered exponentially scaled up versions of R_{max} such as $2R_{max}, 4R_{max}, \ldots 2^k R_{max}$. For every scaling of R_{max}, we correspondingly scaled the tfidf threshold by exponential factor β. To summarize, our rarity metric can be stated as follows:

A term is *rare* if one of two conditions holds - the base condition:

$$LDC(t) < R_{max}, tfidf(t) > R_{th}$$

Or the secondary condition - for $1 \leq l \leq k$:

$$2^{l-1} R_{max} \leq LDC(t) < 2^l R_{max}$$

and

$$tfidf(t) > \beta^k R_{th}$$

If either of these conditions is true, the term is considered rare.

In practice, we set $k = 10$ and $R_{max} = 1000$, to consider all rare terms with a LDC frequency of up to $1,024,000$. We choose a base value of $R_{th} = 0.2$ in our classifier in a similar fashion to T_{th} based on manual inspection across 15 focused topics. We choose β such that $\beta^k = T_{th}/R_{th} = 4/0.2 = 200$. This is to ensure that the rarity metric converges with the popularity metric at an LDC frequency of $2^k R_{max}$. Hence if a term has an LDC frequency greater than this value, and $tfidf(t) > T_{th}$ it gets captured by the popular term classifier.

Given the list of m popular and rare terms extracted for a given topic, given any document d, our feature extraction algorithm will output two $m-$dimensional vectors: (a) the feature vector of $tf(i,d)$ which represents the term frequency of the i^{th} feature term (for $1 \leq i \leq m$) in the document; (b) the $tfidf(i)$ weighting function for each of the m terms. These vectors can be fed to any standard classifier. In practice, we can consider the weighting function for a given term either based on $tf(i,d)$ (no weighting) alone or $tf(i,d) \times tfidf(i)$ (tfidf based weighting) before feeding the vectors in a standard classifier. In our analysis, we use naive-Bayes and SVM as two classifiers in our study.

5. Evaluation

Feature Extraction To evaluate our feature extraction algorithm for a given topic, we generate a training set of candidate documents based on querying a search engine for the topic and extracting the text from the resulting pages. Filtering is applied to this initial set to remove uninformative documents, as mentioned in section 4. The remaining documents are randomly partitioned to generate training and test data sets. In addition a negative test set is generated by extracting 10000 random documents from the English wikipedia corpus.

We evaluated our technique on a variety of different topics that we felt have significance for developing regions. We grouped these into 5 broad categories:

- Actors (Rajinikanth, Amitabh Bachchan, Kamal Haasan)
- Diseases (HIV, Malaria, TB, Diabetes)
- Agriculture (Organic Farming, Water Harvesting, Soya)
- Math (Trigonometry, Calculus)
- Computer Science (Operating Systems, Networking, Compilers)

The number of features kept by the filtering process is shown in Figure 3.

Topic	Original Features	Popular	Rare
amitabh bachchan	6107	75	973
calculus	5848	85	395
compilers	6919	123	793
diabetes	6803	184	545
hiv	7749	217	681
kamal haasan	5246	43	675
malaria	8616	161	844
networking	9922	177	796
operating systems	9796	156	729
organic farming	9315	84	434
rajinikanth	5862	138	1162
soya	7948	87	861
tb	11737	495	1390
tcp	6743	340	1054
trigonometry	5284	105	474
water harvesting	8800	73	404

Figure 3: Popular and rare features extracted by topic

Note the number of features kept by the rare and popular filters is very small relative to the orignal feature counts. The time taken by feature extraction is trivial in comparison to the classification time.

An example of terms computed by the rare filter, from the faculty group of WebKB, and the organic farming group are shown in figure 4. Similar results arise for other topics.

Classification Results We tested 2 classifiers on our problem set, a naive Bayes classifier and an SVM learner; we used the bow [McCallum1996] toolkit to construct and test our classifiers. The models used for the classifiers were generated using the text of the documents as a unigram bag-of-words model. The classifiers were separately trained and tested using a set of words corresponding to the union of the rarity and popularity filters.

Our classifiers were run with their default settings. For libbow, this generates a learner based on a unigram word model. The naive Bayes learner is smoothed by assuming a Dirichlet prior for zero valued features. The SVM learner uses a linear kernel with linear weighting on term frequencies.

Faculty	Organic Farming
eecs	farming
proc.	usda
systems	gardening
m.s.	pesticides
a.g.	organics
u.s.	growers
university	fertilizers

Figure 4: Top terms kept by the rare filter

The results of evaluating our topics were surprising - both the SVM and Bayesian classifier had very high precision the full range of subjects.

For this particular classification task, we found the effectiveness of the Naive Bayes classifiers to be significantly better then our SVM learner when training against the full feature set; the reverse occurs when training against the restricted word set. The effect of filtering terms from the feature set dramatically altered the behavior for our classifiers, in differing manners.

When working on the filtered set, our naive Bayes classifier lost a small amount of recall, and showed better precision when rejecting documents of the negative test set. The classifier exhibited perfect precision in rejecting random documents on 3 of the topics. The most dramatic change was within our math topic set. We suspect the reason for the loss of recall to be related to the distribution of words within the math topic: the number of related but uncommon words for that topic is significantly larger then for the other sets.

Topic	Positive Precision	Negative Precision
cs	0.9637 (0.9477)	0.9900 (1.0000)
agriculture	0.9440 (0.9043)	0.9993 (1.0000)
math	0.9572 (0.8512)	0.9985 (0.9999)
actor	0.9736 (0.9885)	1.0000 (0.9999)
disease	0.9911 (0.9700)	0.9984 (1.0000)

Figure 5: Naive Bayes results, original and with filtering

When run against the filtered set, the SVM learner showed a uniform improvement in recall (acceptance related documents) for all categories and trivial losses in precision (rejecting unrelated documents). The SVM learner operating against the filtered feature sets outperformed all of our other classification attempts by a significant margin.

A beneficial side-effect was noticed while training the reduced feature SVM - the time to train the SVM was drastically reduced when running on the filtered features.

WebKB Evaluation As a base of comparison with related work we evaluated our techniques against the WebKB dataset, though we did not expect to achieve groundbreaking results in this area. The resulting performance was better then anticipated. We followed the procedure used

Topic	Positive Precision	Negative Precision
cs	0.9032 (0.9947)	0.9993 (0.9986)
agriculture	0.9491 (0.9898)	0.9993 (1.0000)
math	0.9419 (0.9808)	0.9992 (0.9984)
actor	0.9761 (0.9915)	0.9997 (0.9993)
disease	0.9837 (0.9931)	0.9997 (0.9990)

Figure 6: SVM results, original and with filtering:

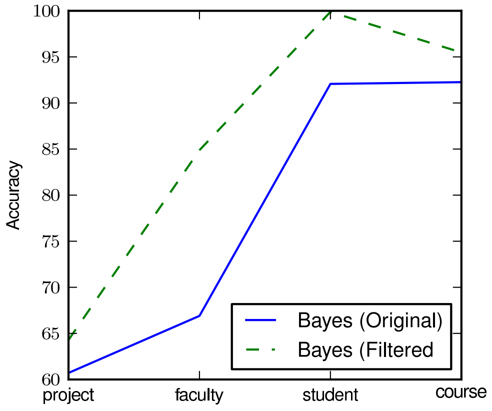

Figure 7: Naive Bayes results across categories

in [Nigam, Lafferty, and McCallum1999], and focused on the joint classification of the reduced set of 4 groups: course, faculty, project and student classes. Once again, we saw differing behavior in our classifiers. For this document set, our naive Bayes learner on the trimmed features outperformed the other classifiers - achieving an aggregate accuracy of 90.7%. Without feature filtering, the Bayes learner achieved only 81.6%. The SVM learners exhibited the reverse behavior - the learner operating on the trimmed set exhibited significantly decreased accuracy.

Upon investigation, we found that our thresholds for this data set had been too aggressive - only 7 words were being kept for the “student” class via the popular filter. The lack of features to work with significantly handicaps the SVM learner. The naive Bayes learner, however, extracts most of the weighting for classification from exactly these rare terms, and hence does not exhibit the same degradation; the removal of spurious terms prevents the classifier from overemphasizing them.

We note that the results for the filtered Bayesian learner are roughly on par with the best techniques we are aware of. Given the relative simplicity of our approach, we feel this is an area worth further study.

SVM (Original)	89.8%
SVM (Filtered)	80.2%
Naive Bayes (Original)	81.7%
Naive Bayes (Filtered)	90.7 %

Figure 8: WebKB classification accuracy

6. Conclusion

We have found that while page classification in general is a hard problem it is not a necessarily difficult problem for all subject areas. We show that when classification tasks are restricted to relatively narrow topics of interest, we can achieve near perfect precision and recall.

We also introduce a novel technique for filtering the terms provided to a classifier, which we show can enhance the effectiveness of both SVM and naive Bayes classifiers for both focused and traditional classification problems.

References

Angelova, R., and Weikum, G. 2006. Graph-based text classification: learn from your neighbors. *ACM SIGIR*.

Baykan, E.; Henzinger, M.; Marian, L.; and Weber, I. 2009. Purely url-based topic classification. *18th International Conference on World Wide Web*.

Chakrabarti, S.; Dom, B.; and Indyk, P. 1998. Enhanced hypertext categorization using hyperlinks. *SIGMOD ’98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*.

Craven, M.; DiPasquo, D.; Freitag, D.; and McCallum, A. 1998. Learning to extract symbolic knowledge from the world wide web. *AAAI Proceedings*.

Domingos, P., and Pazzani, M. 1996. Beyond independence: Conditions for the optimality of the simple bayesian classifier. *Machine Learning* 29:105–112.

Jain, S.; Fall, K.; and Patra, R. 2004. Routing in a delay tolerant network. *ACM SIGCOMM 2004*.

Joachims, T.; Nedellec, C.; and Rouveirol, C. 1998. Text categorization with support vector machines: learning with many relevant. *10th European Conference on Machine Learning*.

Joachims, T. 1998. Making large scale svm learning practical. *Advances in Kernel Methods, Support Vector Learning*.

Jones, K. S. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28:11–21.

Kan, M., and Thi, H. 2005. Fast webpage classification using url features. *ACM CIKM*.

Lewis, D. 1998. Naive (bayes) at forty: The independence assumption in information retrieval. *Lecture Notes in Computer Science*.

McCallum, A. K. 1996. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow.

Nigam, K.; Lafferty, J.; and McCallum, A. 1999. Using maximum entropy for text classification.

Pentland, A.; Fletcher, R.; and Hasson, A. 2004. Daknet: Rethinking connectivity in developing nations. *Computer*.

Qi, X., and Davison, B. 2009. Web page classification: Features and algorithms. *ACM Computing Surveys (CSUR)*.

Seth, A.; Kroeker, D.; Zaharia, M.; and Guo, S. 2006. Low-cost communication for rural internet kiosks using mechanical backhaul. *Mobicom 2006*.

Shih, L., and Karger, D. 2004. Using urls and table layout for web classification tasks. *Proceedings of the 13th international conference on World Wide Web*.

Yang, Y.; Slattery, S.; and Ghani, R. 2002. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*.

Zhang, H. 2004. The optimality of naive bayes. *FLAIRS Conference*.