URL-based Web Page Classification
A New Method for URL-based Web Page Classification Using n-Gram Language Models

Tarek Amr Abdallah and Beatriz de La Iglesia
School of Computing Sciences, University of East Anglia, Norwich Research Park, Norwich, NR4 7TJ, U.K.

Keywords: Language Models, Information Retrieval, Web Classification, Web Mining, Machine Learning.

Abstract: This paper is concerned with the classification of web pages using their Uniform Resource Locators (URLs) only. There is a number of contexts these days in which it is important to have an efficient and reliable classification of a web-page from the URL, without the need to visit the page itself. For example, emails or messages sent in social media may contain URLs and require automatic classification. The URL is very concise, and may be composed of concatenated words so classification with only this information is a very challenging task.
Much of the current research on URL-based classification has achieved reasonable accuracy, but the current methods do not scale very well with large datasets. In this paper, we propose a new solution based on the use of an n-gram language model. Our solution shows good classification performance and is scalable to larger datasets. It also allows us to tackle the problem of classifying new URLs with unseen sub-sequences.

1 INTRODUCTION

During 2010 twitter users sent about 90 million updates every day, as reported by Thomas et al. (Thomas et al., 2011). It is estimated that 25% of those updates contain web-links. Similarly, a huge number of links are carried by the millions of email messages and Facebook updates sent every day. In such context, it is crucial to be able to classify web-pages in real-time using their URLs only, without the need to visit the pages themselves, even if some accuracy is sacrificed for the sake of greater speed of classification. Also, search engines depend mainly on textual data to retrieve on-line resources. However, they are often faced with multimedia content such as videos and images with scarce descriptive tags or surrounding text. Thus, in this context, URL-based classification can be used to decide the categories of such content enhancing the retrieval performance.

Additionally, the classification approach presented here is not limited to URL-based classification tasks only. It can also be adapted for similar problems where there is a need to classify very concise documents with no obvious boundaries between words, e.g. social networks folksonomies.

Unlike documents, URLs are very concise as they are composed of very few words. Usually, words are also concatenated without intermediate punctuations or spaces; for example: *carsales.com* and *vouchercodes.co.uk*. They also contain various abbreviations and domain-specific terms. Therefore, classification requires specific approaches that can deal with the special characteristics of the data under consideration.

2 RELATED WORK

Previous researchers have focused on how to extract features from URLs. Early approaches segmented URLs based on punctuation marks using the resulting terms as the classifier's feature-set (Kan, 2004). Later on, researchers used either statistical or brute-force approaches to further segment URLs beyond the punctuation marks. The non-brute-force approaches used *information content* (Kan, 2004), *dictionary based tokenizes* (Vonitsanou et al., 2011) and *symmetric/non-symmetric sliding windows* (Nicolov and Salvetti, 2007). The brute-force approach, on the other hand, tends to extract all possible sub-strings, *all-grams*, to use them as the classifier's feature-set (Baykan et al., 2009; Baykan et al., 2011; Baykan et al., 2013; Chung et al., 2010). To our knowledge, this is the most successful so far, however, it is ob-

Amr Abdallah T. and de la Iglesia B..
URL-based Web Page Classification - A New Method for URL-based Web Page Classification Using n-Gram Language Models.
DOI: 10.5220/0005030500140021
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2014), pages 14-21
ISBN: 978-989-758-048-2

vious that it does not scale very well. Using such an approach, the resulting datasets in the experiments reported here can go beyond our computational resources and therefore become difficult to store, classify or even to select subset of features from.

The aforementioned classification algorithms are sometimes called batch algorithms, as opposed to online algorithms. In recent research, online learners have been used in URL-based classifications (Ma et al., 2009; Zhao and Hoi, 2013). Nevertheless, they incorporate meta-features, such as WHOIS and geographic information, in addition to the URLs' lexical features. We prefer to limit ourselves here to features found in the URLs only.

Our proposed approach tries to classify URLs without the need to segment them. We borrow the concept of language models from the information retrieval and automatic speech recognition field. We apply a similar approach to that used by Peng et. al. to classify Japanese and Chinese documents (Peng et al., 2003). They used an *n-Gram Language Model (LM)* in order to classify textual data without the need for segmenting into separate terms. These two east-Asian languages are similar to URLs in the sense that spaces between words are absent so we hypothesise that a similar approach can work for the URL classification problem. We have adapted the model used by Peng et. al to be used with URLs, given their format and punctuations. Furthermore, we made use of the *Linked Dependence Assumption* to relax the model's independence assumption and to improve its performance. We further expand on this in section 3.2.

In the next section we are going to explain the n-gram Language Model and its use for document classification. In section 4 we give more details on the dataset used, and the experiments done. Then, we present our results in sections 5. Finally, we conclude our findings and offer suggestions for future researchers in the last section.

3 THE N-GRAM LANGUAGE MODEL

Let us assume we have a set of documents $D = \{d_1, d_2, ..., d_m\}$, and a set of classes $C = \{c_1, c_2, ..., c_k\}$, where each document is classified as member of one of these classes. For any document, d_i, the probability that it belongs to class c_j, can be represented as $Pr(c_j/d_i)$ and using Bayes rules (Zhai and Lafferty, 2001; Peng et al., 2003), this probability is calculated by:

$$Pr(c_j/d_i) = \frac{Pr(d_i/c_j) * Pr(c_j)}{Pr(d_i)} \quad (1)$$

The term $Pr(d_i)$ is constant for all documents. The term $Pr(c_j)$ can represent the distribution of class j in the training set. A uniform class distribution can also be assumed, so we end up with the term $Pr(d_i/c_j)$ only (Grau et al., 2004). For a document d_i, that is composed of a sequence of words $w_1, w_2, ..., w_L$, $Pr(d_i/c_j)$ it is expressed as follows: $Pr(w_1, w_2, ...w_L/c_j)$. We are going to write it as $Pr_{c_j}(w_1, w_2, ...w_L)$ for simplicity.

$Pr_{c_j}(w_1, w_2, ...w_L)$ is the likelihood that $w_1, w_2, ..., w_L$ occurs in c_j. This can be calculated as shown in equation 2.

$$\begin{aligned} & Pr_{c_j}(w_1, w_2, .., w_{L-1}, w_L) \\ & = Pr_{c_j}(w_1) * Pr_{c_j}(w_2/w_1) \\ & *... * Pr_{c_j}(w_L/w_{L-1}, .., w_1) \\ & = \Pi_{i=1}^{L} Pr_{c_j}(w_i/w_{i-1}, w_{i-2}, ..., w_1) \end{aligned} \quad (2)$$

Nevertheless, in practice, the above dependency is relaxed and it is assumed that each word w_i is only dependent on the previous $n-1$ words (Peng et al., 2003). Hence, equation 2 is transformed to the following equation:

$$\begin{aligned} & Pr_{c_j}(w_1, w_2, ...w_L) \\ & = \Pi_{i=1}^{L} Pr_{c_j}(w_i/w_{i-1}, w_{i-2}, ..., w_{i-n+1}) \end{aligned} \quad (3)$$

The n-gram model is the probability distribution of sequences of length n, given the training data (Manning and Schütze, 1999). Therefore, $Pr_{c_j}(w_1, w_2, ...w_L)$ is referred to as the n-gram language model approximation for class c_j. Now, from the training set and for each class, the n-gram probabilities are calculated using the maximum likelihood estimation (MLE) shown in equation 4 (Chen and Goodman, 1996):

$$\begin{aligned} Pr_{c_j}(wi/w_{i-n+1}^{i-1}) &= \frac{Pr(w_{i-n+1}^{i})}{Pr(w_{i-n+1}^{i-1})} \\ &= \frac{count(w_{i-n+1}^{i})/N_w}{count(w_{i-n+1}^{i-1})/N_w} \\ &= \frac{count(w_{i-n+1}^{i})}{count(w_{i-n+1}^{i-1})} \end{aligned} \quad (4)$$

where N_w is the total number of words, and w_{i-n+1}^{i} is the string formed of the 'n' consecutive words between w_{i-n+1} and w_i. We are proposing to use the n-Gram Language model for URL-based classification. However, in our case, we will use characters instead of words as a basis of the language model. We construct a separate LM for each class of URLs

as follows. The above probabilities are calculated for each class in the training set by counting the number of times all sub-strings of lengths n and $n-1$ occur in the member URLs of that class. For example, suppose we have the following strings as members of class c_j, {'ABCDE','ABC','CDE' }. In a 3-gram LM, for class c_j we will store all sub-strings of length 3 and those of length 2, along with their counts, as shown in table 1.

Table 1: Sample data-structure for 3-gram LM counts.

3-grams	('ABC': 2), ('BCD': 1), ('CDE': 2)
2-grams	('AB': 2), ('BC': 2), ('CD': 2), ('DE': 2)

Counts in table 1 are acquired during the training phase. Then in the testing phase, URLs are converted into n-grams, and for each n-gram, its probability is calculated using equation 4. A new URL, URL_i, is classified as member of class c_j, if the language model of c_j maximizes equation 1, i.e. maximizes $Pr(c_j/URL_i)$.

3.1 Dealing with Unseen n-Grams

The maximum likelihood in equation 4 can be zero for n-grams not seen in the training set. Therefore, smoothing is used to deal with the problem by assigning non-zero counts to unseen n-grams. Laplace smoothing is one of the simplest approaches (Chen and Goodman, 1996), calculated as follows:

$$Pr_{c_j}(wi/w_{i-n+1}^{i-1}) = \frac{count(w_{i-n+1}^{i})+1}{count(w_{i-n+1}^{i-1})+V} \quad (5)$$

In equation 5, the count is increased by 1 in the numerator, and by V in the denominator, where V represents the number of unique sequences of length $n-1$ found in the training set. By using this, we are effectively lowering the count of the non-zero sequences and assigning a discounted value to the unseen sequences (Jurafsky and Martin, 2000). Both 1 and V can be multiplied by a coefficient γ in order to control the amount of the probability mass to be re-assigned to the unseen sequences. There are other more sophisticated smoothing techniques that could be applied including Witten-Bell discounting (Witten and Bell, 1991) and Good Turing discounting (Good, 1953).

3.2 Linked Dependence Assumption

In the n-gram LM, in order to move from equation 2 to equation 3, we need to assume that the probability of w_i depends only on that of the previous $n-1$ terms. Similarly, in the uni-gram LM, all terms are assumed to be totally independent, i.e. it is equivalent to a bag of words approach. Although, increasing the value of n relaxes the *independence assumption*, it is still a strong assumption to make. Cooper (Cooper, 1995), points out the *linked dependence assumption* (LDA) as a weaker alternative assumption. Lavrenko (Lavrenko, 2009) explained the *linked dependence* as follows. Consider the case of a two words vocabulary, $V = \{a,b\}$. In the case of two classes, c_1 and c_2, and under the *independence assumption*, $Pr_{c1}(a,b) = Pr_{c1}(a) * Pr_{c1}(b)$. Similarly $Pr_{c2}(a,b)$ is the product of $Pr_{c2}(a)$ and $Pr_{c2}(b)$. Otherwise, when terms are assumed to be dependent, $Pr_{c1}(a,b)$ and $Pr_{c1}(a,b)$ can be expressed as follows:

$$Pr_{c_j}(a,b) = K_{c_j} * Pr_{c_j}(a) * Pr_{c_i}(b) \quad (6)$$

where K_{c_j} measures the dependence of the terms in class c_j. Terms are positively correlated if $K_{c_j} > 1$, and they are negatively correlated if $K_{c_j} < 1$. As mentioned earlier, with the independence assumption, K_{c_j} is equal to 1. Now, in Cooper's LDA, K_{c_j} is not assumed to be equal to 1, however it is assumed to be the same for all classes, i.e. $K_{c_1} = K_{c_2} = K_{c_j} = K$

Accordingly, the value of K might not be needed if we try to maximize the log-likelihood ratio of relevance of $Pr(c_j/d_i)$ divided by $Pr(\bar{c}_j/d_i)$, rather than $Pr(c_j/d_i)$ as in equation 1. $Pr(\bar{c}_j/d_i)$ is the posterior probability of all other classes except c_j. This is similar to the approach used in the *binary independence model* (BIM) (Robertson and Jones, 1976; Sparck Jones et al., 2000). Similarly, in the case of using Language Models for spam detection, Terra created two models for ham and spam messages (Terra, 2005), and a message was considered to be spam if its log-likelihood odds ratio exceeded a certain ratio. Hence, the equation of our proposed classifier will look as follows.

$$\begin{aligned} logLL_{c_j} &= log(\frac{Pr(c_j/d_i)}{Pr(\bar{c}_j/d_i)}) \quad (7)\\ &= log(\frac{Pr(d_i/c_j) * Pr(c_j)}{Pr(d_i/\bar{c}_j) * Pr(\bar{c}_j}) \\ &= \Sigma_{i=1}^{L} log(\frac{Pr_{c_j}(w_{i-n+1}^{i})}{Pr_{\bar{c}_j}(w_{i-n+1}^{i})}) + log(\frac{Pr(c_j)}{Pr(\bar{c}_j)}) \end{aligned}$$

A new URL, URL_i, is classified as member of class c_j, if the language model of c_j maximizes equation 7, i.e. maximizes the $logLL_{c_j}$. Hereafter, we refer to this variation of the n-gram LM as *Log-likelihood Odds* (LLO) model. It is worth mentioning that the use of logarithmic scale also helps in preventing decimal point overflow during the implementation.

Table 2: Comparing $F_1 - measure$ for the WebKB dataset. Results in first 3 rows are from (Kan, 2004) using SVM^{light}, the last two rows are using the proposed n-gram Language Model ($\gamma = 0.0062$). IC, FST and LLO stand for information content reduction, title token-based finite state transducer, and Log-likelihood Odds respectively. All F_1 values are multiplied by 100.

Classifier	Course	Faculty	Project	Student	Macro Avg.
Terms	13.5	23.4	35.6	15.8	22.1
IC	50.2	31.8	35.0	15.7	33.2
FST	52.7	31.5	36.3	15.6	34.0
All-Grams	78	**75**	50	**63**	**66.5**
4-gram LM/LLO	**83.6**	40.2	**53.7**	59.4	59.25

4 EXPERIMENTS AND DATASETS

Two datasets are used here. WebKB corpus is commonly used for web classification (e.g. (Slattery and Craven, 1998)). It contains pages collected from the computer science departments in 4 universities. We employed the same subset of the dataset used in previous research, to be able to compare our results to them (Kan, 2004; Kan and Thi, 2005; Baykan et al., 2009). The subset contains 4,167 pages. In a similar fashion to previous research, we also used the same training and test-sets and a *leave-one-university-out cross-validation* for the WebKB URLs (Kan, 2004).

In addition to WebKB, we also used the categorized web pages from DMOZ, which was historically known as the Open Directory Project (ODP). In (Baykan et al., 2009), they selected 15 topics from DMOZ categories, 1,000 URLs were put aside for testing, and the remaining URLs were used to create 15 balanced training sets for their 15 binary classifiers. For the sake of comparison, we calculated the precision, recall and F-measure for this dataset in the same fashion as explained in (Baykan et al., 2008).

5 RESULTS

5.1 Results for the Primary Dataset

(Kan, 2004) achieved an average $F_1 - measure$ of 22.1% for the WebKB dataset using punctuation-based (terms) approach. They then tried the *information content* (IC) reduction and *title token-based finite state transducer* (FST) to further segment URL terms and expand abbreviations, achieving an average $F_1 - measure$ of 33.2% and 34% respectively. For the same dataset, the proposed n-gram LM classifier achieved an average $F_1 - measure$ of 51.4%, where $n = 4$ and $\gamma = 0.0062$. The *log-likelihood odds* (LLO) variation of the same LM increased the average $F_1 - measure$ to 59.25%. Detailed results are shown in table 2.

In later research, (Kan and Thi, 2005) tried additional feature extraction methods, achieving the highest $F_1 - measure$ of 52.5%. For the same dataset, (Baykan et al., 2009) and (Baykan et al., 2011) reported $F_1 - measure$ of 66.5% using the all-gram approach. It is clear that the classification performance of the n-gram LM for this dataset is better than all previous approaches except for all-grams. Nevertheless, the difference between results for all-grams and that of the n-Gram LM are not statistically significant, p=0.5. Furthermore, it is worth noting that the n-gram LM uses only 4-grams and requires about 0.04% of the storage and memory needed for the all-grams approach. More discussion on the scalability of the n-gram LM is included in section 6.

5.2 Results for the Secondary Dataset

The results for DMOZ dataset are shown in table 3. The best results for the n-gram LM were achieved using 7-grams and $\gamma = 0.004$. The results for the previous research using SVM and all-gram features (all 4,5,6,7 and 8-grams) (Baykan et al., 2009), are also shown in the table. The performance of the n-gram LM is marginally better, however the statistical analysis of the results confirms that there is no statistical significance between the accuracy of the two approaches. Again, for some classes, the n-Gram LM requires less than 0.001% of the memory and storage needed by the all-gram approach. The scalability of the n-gram LM is discussed in section 6.

5.3 n-Gram LM Parameter Experimentation

Two main parameters play an important role in our n-gram LM results:

1. The order of n in the n-gram LM.
2. The value of γ in Laplace smoothing.

Table 3: Comparing the F-measure of the n-Gram LM and SVM (all-gram features) classifiers for DMOZ dataset. All F_1 values are multiplied by 100.

Topic	SVM all-gram	n-Gram LM/LLO
Adult	87.6%	87.58%
Arts	81.9%	82.03%
Business	82.9%	82.71%
Computers	82.5%	82.79%
Games	86.7%	86.43%
Health	82.4%	82.49%
Home	81%	81.13%
Kids	80%	81.09%
News	80.1%	79.01%
Recreation	79.7%	80.22%
Reference	84.4%	83.37%
Science	80.1%	82.52%
Shopping	83.1%	82.48%
Society	80.2%	81.66%
Sports	84%	85.30%
Average	82.44%	82.72%

There is a trade-off between smaller and larger values of n. Higher values of n imply more scarce data and a higher number of n-grams in the testing phase that have not been seen during the training phase. On the other hand, for a lower value of n, it is harder for the model to capture the character dependencies (Peng et al., 2003). The quantity of unseen n-grams in the testing phase is also dependent on the class distributions and the homogeneity of the class vocabularies. Classes with more samples have more chance to cover more n-gram vocabulary.

In this context, smoothing is needed to estimate the likelihood of unseen n-grams. The value of γ controls the amount of probability mass that is to be discounted from seen n-grams and re-assigned to the unseen ones. The higher the value of γ the higher the probability mass being assigned to unseen n-grams.

Figure 1 shows the variation of the F-measure with the value of n the in n-gram LM, for the different class labels in the WebKB dataset. The macro-average F-measure is also shown in the figure. It is clear that the best results are achieved at n=4.

Similarly, the effect of the smoothing parameter (γ) is shown in figure 2. Figure 2 also shows that relaxing the model's *independence assumption*, by using the *Log-likelihood Odds* model, results in better performance, and more immunity to the variations of the smoothing parameter.

When the model encounters a high percentage of n-grams that were never seen during the training phase, the precision of the model is affected. Smoothing, on the other hand, tries to compensate this effect by moving some of the probability mass to the unseen n-grams. As stated earlier, the amount of the probability mass assigned to the unseen n-grams is controlled by the value of γ.

In figure 3, we can see the correlation between the precision and the percentage of seen n-grams for the different classes. It is also clear that the correlation gets stronger with lower values of γ. For the shown models, the Pearson correlation coefficients for the precision values with the percentages of seen n-grams are 0.51, 0.65 and 0.74 for $\gamma = 1$, 0.1 and 0.01 respectively.

6 N-GRAM LM SCALABILITY

The storage size needed for the n-gram LM is a function of the number of n-grams and classes we have, while for the all-grams approach used by Baykan et al. (Baykan et al., 2009), the storage requirements are a function of the number of URLs in the training set as well as the different orders of 'n' used in the all-grams. This means that in the n-gram LM the memory and storage requirements can be 100,000 times less than that needed by the conventional approaches. This reduction was shown, during our tests, to also have a big impact on the classification processing time.

Let us use any of the binary-classifiers used in DMOZ dataset to explain this in more details. We have about 100,000 URLs in the 'Sports' category, thus as shown in Baykan et al. (Baykan et al., 2009), we will build a balanced training-set of positive and negative cases of about 200,000 URLs.

As we have seen in equations 4 and 5, for an n-gram language model we need to store the counts of n-grams and (n-1)-grams for each class. Since we can achieve slightly better results than Baykan et al. (Baykan et al., 2009) with 'n=7', we will do our calculations based on the 7-gram LM here. The number of 7-grams in the positive and negative classes are 746,024 and 1,037,419 respectively, while the number of 6-grams for the same two classes are 568,162 and 795,192. Thus the total storage needed is the summation of the above 4 values, i.e. 3,146,797

For the approach used by Baykan et al. (Baykan et al., 2009), we need to construct a matrix of all features and training-data records. The features in this case will be the all-grams, i.e. 4, 5, 6, 7 and 8-grams, and the training-data records are the 200,000 URLs in the training-set. This matrix is to be used by a Naive Bayes or SVM classifiers later on. The counts for the 4, 5, 6, 7 and 8-grams are 222,649, 684,432, 1,198,689, 1,628,422, 2,008,153 respectively. Thus, the total number of features is the sum-

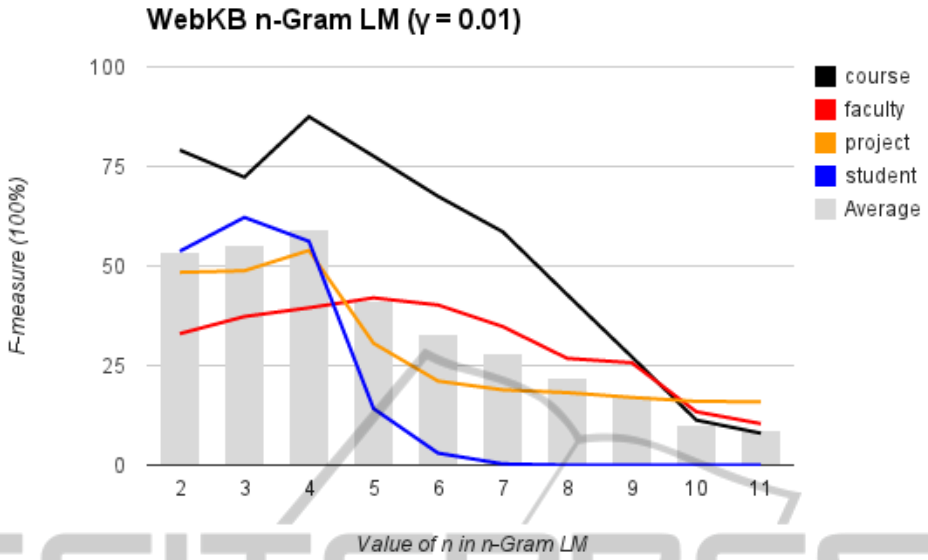

Figure 1: Variations of F-measures with n.

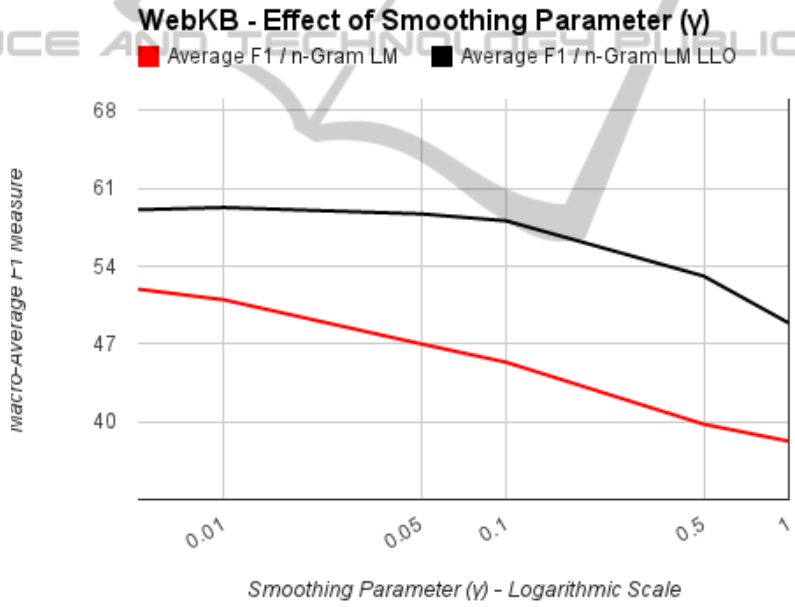

Figure 2: Variations of F-measures with γ.

mation of the above 5 values, i.e. 5,742,345. Given that there are about 200,000 URLs in the training-set, the total size of the matrix will be the product of the above 2 numbers, $5,742,345 * 200,000$, which is 1,148,469,000,000.

As we can see in the above example, the memory and storage requirements for the n-gram LM is 1:364,964 ($\approx 0.0003\%$) of that needed for the conventional approaches. Similarly, even for a small datasets such as WebKB, the memory needed for n-gram LM is about 1:2600 ($\approx 0.04\%$) of that needed for the all-grams approach.

As we have discussed earlier, such reduction in storage and processing requirements for the n-Gram LM, does not impact negatively on its classification performance compared the the previous classification approaches.

7 CONCLUSIONS

Here we have presented a new LM approach for URL classification that cuts down on the number of fea-

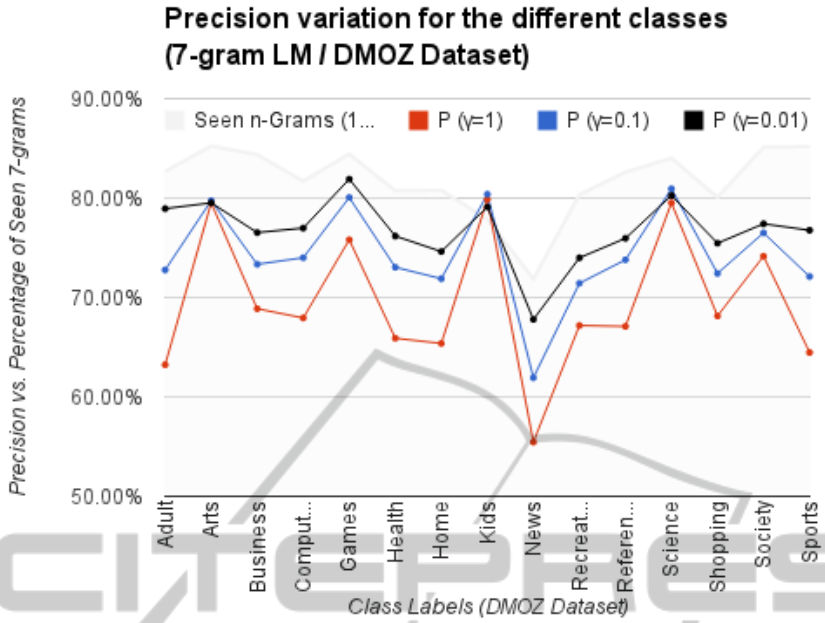

Figure 3: Variations of Precision with classes and with the smoothing parameter, γ.

tures, and therefore, the storage and processing requirements, and still manages to achieve comparable levels of performance. Our experiments show that the n-gram LM approach with very basic smoothing is offering some significant improvements for classification performance in some cases or at least equal performance over other methods such as *terms* or *all-grams* used with NB and SVM classifiers.

The n-gram LM requires less processing power compared to all-gram. For some cases the proposed model required less that 0.001 % of the storage and processing power needed by the previous methods.

Our method has application to real world URL classification, an important emerging problem. We have tested it on a large dataset (some classes of DMOZ dataset have more that 200,000 URLs) as well as on the WebKB dataset. We have also performed parameter experimentation to establish the importance of parameters in the new LM.

As further work, we believe that more sophisticated smoothing methods and interpolating multiple n-gram models, with different values of n, could improve the performance of the LM model. Thus, we propose to continue our research in that direction.

REFERENCES

Baykan, E., Henzinger, M., Marian, L., and Weber, I. (2009). Purely url-based topic classification. In *Proceedings of the 18th international conference on World wide web*, pages 1109–1110. ACM.

Baykan, E., Henzinger, M., and Weber, I. (2008). Web page language identification based on urls. *Proceedings of the VLDB Endowment*, 1(1):176–187.

Baykan, E., Henzinger, M., and Weber, I. (2013). A comprehensive study of techniques for url-based web page language classification. *ACM Transactions on the Web (TWEB)*, 7(1):3.

Baykan, E., Marian, L., Henzinger, M., and Weber, I. (2011). A comprehensive study of features and algorithms for url-based topic classification. *ACM Transactions on the Web (TWEB)*, 5(3):15.

Chen, S. F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 310–318. Association for Computational Linguistics.

Chung, Y., Toyoda, M., and Kitsugeregawa, M. (2010). Topic classification of spam host based on urls. In *Proceedings of the Forum on Data Engineering and Information Management (DEIM)*.

Cooper, W. S. (1995). Some inconsistencies and misidentified modeling assumptions in probabilistic information retrieval. *ACM Transactions on Information Systems (TOIS)*, 13(1):100–111.

Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, 40(3-4):237–264.

Grau, S., Sanchis, E., Castro, M. J., and Vilar, D. (2004). Dialogue act classification using a bayesian approach. In *9th Conference Speech and Computer*.

Jurafsky, D. and Martin, J. (2000). *Speech & Language Processing*. Pearson Education India.

Kan, M. (2004). Web page classification without the web page. In *Proceedings of the 13th international World*

Wide Web conference on Alternate track papers & posters, pages 262–263. ACM.

Kan, M. and Thi, H. (2005). Fast webpage classification using url features. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 325–326. ACM.

Lavrenko, V. (2009). *A generative theory of relevance*, volume 26. Springer.

Ma, J., Saul, L. K., Savage, S., and Voelker, G. M. (2009). Identifying suspicious urls: an application of large-scale online learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 681–688. ACM.

Manning, C. D. and Schütze, H. (1999). *Foundations of statistical natural language processing*, volume 999. MIT Press.

Nicolov, N. and Salvetti, F. (2007). Efficient spam analysis for weblogs through url segmentation. *Amsterdam studies in the theory and history if linguistic science. Series 4*, 292:125.

Peng, F., Huang, X., Schuurmans, D., and Wang, S. (2003). Text classification in asian languages without word segmentation. In *Proceedings of the sixth international workshop on Information retrieval with Asian languages-Volume 11*, pages 41–48. Association for Computational Linguistics.

Robertson, S. E. and Jones, K. S. (1976). Relevance weighting of search terms. *Journal of the American Society for Information science*, 27(3):129–146.

Slattery, S. and Craven, M. (1998). Combining statistical and relational methods for learning in hypertext domains. In *Inductive Logic Programming*, pages 38–52. Springer.

Sparck Jones, K., Walker, S., and Robertson, S. E. (2000). A probabilistic model of information retrieval: development and comparative experiments: Part 1. *Information Processing & Management*, 36(6):779–808.

Terra, E. (2005). Simple language models for spam detection. In *TREC*.

Thomas, K., Grier, C., Ma, J., Paxson, V., and Song, D. (2011). Design and evaluation of a real-time url spam filtering service. In *Security and Privacy (SP), 2011 IEEE Symposium on*, pages 447–462. IEEE.

Vonitsanou, M., Kozanidis, L., and Stamou, S. (2011). Keywords identification within greek urls. *Polibits*, (43):75–80.

Witten, I. H. and Bell, T. C. (1991). The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *Information Theory, IEEE Transactions on*, 37(4):1085–1094.

Zhai, C. and Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342. ACM.

Zhao, P. and Hoi, S. C. (2013). Cost-sensitive online active learning with application to malicious url detection. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 919–927. ACM.