DIST
Distributed Intelligent
Systems and Technologies
2-4 July 2012 • St. Petersburg, Russia

USAGE OF THE “BAG OF WORDS MODEL” FOR URL

Roman Hujer
Czech Technical University in Prague
roman.hujer@fsv.cvut.cz

ABSTRACT

In this paper is without a proof supposed that exists a classification of URLs based on their semantics. More about this topic can be found in the paper "Patterns in URL"[RHSZ]. This idea comes from the intuitive human tendency to classify URLs in the web pages to important ones and the unimportant ones.

Nowadays this importance is neglected by the Internet search engines. If there was a classification assigning each class with an importance, the search results could have been more accurate. The importance can be implemented as an unequal pagerank distribution. Example of a simple classification is shown in tab1. Li Cun-He and Lv Ke-qiang with help of that classification improved the Nutch search engine.

Classification can be built with a help of patterns. But patterns in URL are problematic to find. The aim of this article is to explore one method of discovering patterns.

Main idea of this experiment is to use the Bag of words method on URL strings. As an input will be taken a list with a large quantity of URLs. These URLs will be used to create one bag of words along with quantities (frequencies) of these words. It is assumed that words with high frequencies might be patterns in URL or might suggest some possibilities of classification.

Table 1: Simple hyperlink classification consisting of four classes by Cun-He, Li; Ke-qiang, Lv. Each class has assigned by the weight factor [CCLS].

	Is NOT a part of the navigation	Is a part of the navigation
Target is within the same domain	Recommend-ing inner link: $\beta_1=5$	Other inner link: $\beta_2=1$
Target is NOT within the same domain	Recommend-ing outer link: $\beta_3=6$	Other outer link: $\beta_4=1$

INTRODUCTION

There shall be also different approach to creating the classification. I believe that URL itself contains more information than just a locator to a resource. From the syntax of an URL could be extracted an information, which do URL contain aside the locator.

There could be found certain sub-strings that are repeating in URLs. But not literary. In certain patterns. I want to search for these patterns. Identify them, describe them and quantify their frequency.

The patterns have another interesting consequence. Internet contains a lot of resources which cannot be accessed from another resource via hyperlink, so called Deep web[WREX]. You can retrieve them only when you know exact URL. Also there are resources, which are accessible in standard ways, but they could be also accessible in different ways. And it could be beneficial to access them in another way. Pattern could be helpful in accessing the resource in other way. That means a pattern could help us discover previously unknown resources.

The simplest example of a pattern is the numeric pattern. For example URL: http://www.mybook.com/chapter3.html contains an instance of the numeric pattern; number 3. Change of the pattern produce another URL:
http://www.mybook.com/chapter2.html .
This paper describes search for another patterns using the Bag of Words model.

APPLICATION OF A URL CLASSIFICATION IN DISTRIBUTED SYSTEMS

The findings about the patters and URL classification will be also useful in the field of distributed systems. Among the basic requirements on node in distributed systems is that any node can any time get offline or online without a previous notice and the system as a whole would not be affected by that.

In the situation where nodes addressing each other with an URL we can use the pattern in reversed order. All the identifications of nodes will be created to accord to previously defined pattern. When a new node appears in the system the other nodes will discover it without knowing its URL just based on the knowledge of the pattern.

Going even further, the node identifier does not have to be an URL. Generally it can be any string with a semantics. It can be an IP address. It can be used in a such distant discipline as RoboCup *RoboCup is an international robotics competition. The aim is to promote robotics and AI research* [wiki]. In an example of RoboCup the new player in a team, identified by an IP address can be found because his IP address matches the pattern of the team. There are also simpler methods to do the identification. This example shall only demonstrate application of a pattern.

RESEARCH QUESTIONS

- Can be the Bag of words used against URL as input document?
- Can be the resulting bag of words used as a help for URL classification?
- Can be the resulting bag of words used as a source of new patterns?

HYPOTHESIS

- The Bag of words model can be used against list of URLs.
- The Bag of words will be beneficial for discovering patterns in URL and will help with URL classification.

BAG OF WORDS MODEL

So called bag-of-words model is used in natural language processing.

A document is typically stored as sequence of characters. There are methods how to transform document into an atomic structure. Result of application of these methods is a grammar-less, unordered set (bag) of atomic components (words).[B40]

Then bag of words could be used for several kinds of lexical(Bayesian) analysis. For example to measure distance between two documents. To be able to use the bag of words model to URLs a transformation method has to be defined. In order to transform document (=URL) into atomic components identification of a separator character is necessary. In most natural languages is the separator space. This is not true for URL string. In URL is necessary to select more symbols as the separators. Based on empiric evaluation following set of characters was selected as set of separators S : {',',' ','.','/','-','_','?','+','='}

INPUT DATA

For the analysis input data have to be collected first, a list of links. Due to need of large amount of hyperlinks an automated way of their collection was employed. Variety of hyperlinks was also desired. So the traditional crawler, looping over a page and collecting all URLs that it could find, wouldn't suit the purpose. So a web-service provided by digg.com was used.

With this approach various enough data will be obtained. These data already carry some signs of behavior. Thus the input data are not random.

Shall be the input data random or not? Different input data will produce different frequency of pattern occurrence. It is impossible to search through the whole Internet. Certain subset always have to be selected. The preselected subset is better.

Before constructing a Bag of words input data were cleaned from duplicate URLs. Technically that required to sort data alphabetically first. This sorting has no impact to a Bag of words construction.

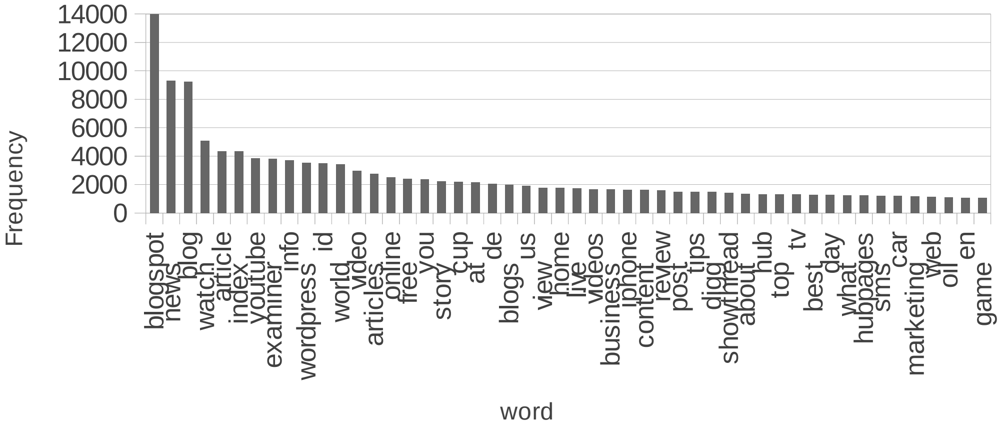

Figure 1: First 50 BOW words after filtering

URL SEPARATION METHOD

Document transformation method into atomic components is obvious. For each document is created a set of non-atomic components NA_1 by dividing it into sub-strings at the places of separator S_1 from S. NA_1 means non-atomic components after first iteration. First component from NA_1 is NA_{11}. Third component from the second iteration is NA_{23}.

Then separation with the separator S_2 is applied to every component from NA_1, resulting into NA_2. Method is applied iteratively for each separator S_1 ... S_n resulting into set NA_n. Set NA_n consist only of atomic components. Bag of words is created by counting frequencies for each component in set Na_n.

The only issue with this approach is to decide what will be the document; input of the separation process. It was found that taking the whole URL as an input for the separation does not produce good results. In the next paragraph will be described filtering of the bag of words. Separation URL in one step posses issues in the filtering afterward, because relevance of a word for identifying a pattern may differ by their position in the URL.

There are several ways how to decompose URL. In this article will be preferred the Pythonic way:[RFC3986]

```
<scheme>://<netloc>/<path>?<query>#<fragment>
```

For example if word "de" is a part of netloc(network location)[rfc1808], it is a country code (top level domain) of Germany, which is not relevant for pattern search. In turn if word "de" is located in another part of URL (path, query) it might be language code, which is a pattern. This example justifies separate bag of words for specific parts of a URL.

Table 2: Stemization ex. - filters comparison

	Original word	Improved English	Porter
1	news	news	new
2	US	US	U
3	this	this	thi
4	pays	pay	pai
5	communities	communiti	commun
6	Asus	Asus	Asu
7	yes	yes	ye
8	wing's	wing	wing'
9	bus	bus	bu
10	1950s	1950s	1950
11	42mbps	42mbps	42mbp
12	ps3s	ps3s	ps3
13	chris	chris	chri
14	sms	sms	sm
15	p5fB9Gk0Ds	p5fB9Gk0Ds	p5fB9Gk0D

This example clearly demonstrates that separation in one step against the whole URL causes semantic loss. Thus is necessary to decompose each URL to netloc, path, query and fragment and apply the separation on each part.

FILTERING OF THE BAG OF WORDS

In the bag of words are also included words which serve only for technical purposes. Typically it is a string identifying the top level domain, file-type identification or other. These words in bag are not relevant for searching of a pattern, thus they have to be filtered out.

Also auxiliary words, which don't carry meaning, such as articles, prepositions, conjugations or pronouns shall be left out of examination. Characteristics of the “Numeric pattern” were already described in another article [RH10]. So it will be kept out of scope of this paper.

STEMIZATION FILTER

After observing the generated bag of words was found that the bag contains many words in plural which duplicate the words in singular. Plurals do not have another semantics, only syntax. Thus it was decided to apply a filter, which joins the singular and plural form. Majority of observed duplicates were in English language. For the purpose of joining duplicate entries in the bag of words was selected stemization filter.

A stemming algorithm (or stemmer) is a process for removing the commoner morphological and inflexional endings from words [PrStW].

There was still question if is for this task more suitable the original Porter algorithm or the improved English algorithm? From a set of 1695 URLs was generated a bag of 6135 words. In order to reduce size of this set were selected only words ending with letter "s"; suspected plurals in English. This reduced set is just an approximation. Bag of the s-ending words contains 1134 of them. The classic Porter stemmer changed 1092 of the s-ending words, whereas the improved filter changed only 1020 of them. In total the result differed in 98 cases.

Table 3: Porter stemmer improved - bad results

	Original word	**Improved English**
1	goods	good
2	articles	articl
3	business	busi
4	reuters	reuter
5	movies	movi
6	goes	goe
7	james	jame
8	texas	texa
9	ios	io
10	miscellaneous	miscellan
11	regulators	regul
12	machines	machin
13	animals	anim
14	economics	econom
15	Windows	Window

Result is obvious. In most cases improved algorithm gives better results than the original.

Is necessary to stress out that majority of differences against expectation are not errors in the algorithm. Only the algorithm was created for different purpose. More unexpected results can be seen in tab3.

For many words the stemization does not give the usable results. So apply this filter in all cases would make more bad than good. Thus is necessary to find another way. Filter will be applied only in cases when the resulting word already exists in the bag. This approach will still produce some errors. The first kind of that error is word after stemization will match to another word with different semantics and those entries will be (incorrectly) joined. This is illustrated by word “goods” (1), which will be joined with word “good” also present in the bag. Similarly can happen that stemmed word matches to a random combination of letters. This is more probable with short words (e.g. ios).

DATA AND TECHNICAL CONSTRAINTS

Size of linklist M was 158713 for this experiment. (Size of text file: 10,88 MB.)
Processing linklist into the bag of words took 8 hrs and 15 minutes.
For saving of the Bag of words was used sqlite3 database. (Resulting BOW database: 3,6 MB.)
It was found 183727 of unique words.
The size of the BOW is 1348945 non-unique words.
Average frequency: 7,4
Count of the words with frequency >1000: 105
Count of the words with frequency >100: 1348
Count of the words with frequency >10: 12111

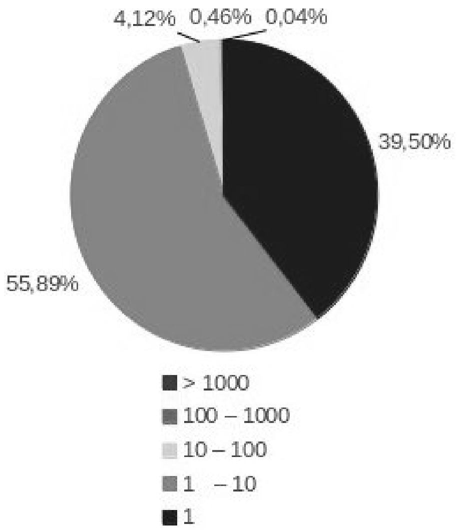

Figure 2: Frequencies by word count. Obviously frequencies over 10 are very rare, which points out significance of such words.

ANALYZING THE CONTENT OF THE BAG OF WORDS

In the bag of words it will be focused on words with high frequencies, because these words could be more likely patterns or support the URL classification. Within these high-frequency components could be identified several categories.

Full patterns

Table 4: Language code pattern

word	path count	notes
en	895	
de	1111	Word 'de' might be also definite article.
es	70	
it	1131	From the number is obvious that only few of these mean the italian language.
fr	102	
jp	11	
ru	6	
zh	10	

Table 5: Month in year pattern

word	netloc count	path count
may	0	325
jun	0	322

First there could be found full patterns. For those makes sense exchanging the pattern occurrence for another occurrence and ask if newly created link leads to another resource. Eg: language code or month in writing.

Content type patterns

In results also can be found words with high frequencies which together create a class of a content type. Words like “article” or review generate class of text content type. Words like “video”, “music” or “photo” generate a multimedia content class. There is also group of words which point out some content type, but do not generate an uniform class.

In spite of the fact that operations with the pattern instance does not have to make sense, content type patterns too, can be useful for finding classification.

Content type patterns appear to generate class itself. But it can be also assumed that there is a transformation for change the pattern, which will produce meaningful results. Only this transformation is unknown so far.

Table 8: Class: Text content

word	netloc count	path count
news	1394	7695
blog	2337	6770
blogs	686	1286
article	51	4149
articles	361	2394
story	7	2200
stories	6	423
review	98	1500
guide	37	516
interview	0	297

Table 7:Class: Multimedia content

word	netloc count	path count
videos	21	1643
video	0	227
movie	47	765
music	53	757
photo	53	546
photos	18	779
pictures	19	414
picture	2	132
streaming	36	344
gallery	28	396

Table 6: Topic group

word	netloc count	path count
marketing	86	1088
oil	8	1085
game	64	989
health	85	795
money	156	723
time	144	726
loans	48	780
games	124	690
sports	234	561
social	29	714
security	61	321

Table 9: Class: Other content

word	netloc count	path count
content	47	1424
download	18	670
archiv	5	614
detail	0	353
apps	14	342

Table 10: Other category

word	netloc count	path count
about	191	1138
product	5	562
technology	11	518
forum	212	814
forums	108	479
help	13	331

Table 11: Products of the Urchin Tracker Module

word	query count	non-match
utm	1509	4
source	657	212
medium	441	23
campaign	431	41
term	51	1
content	758	245

Topic group

Some words with high frequencies do not imply directly content type, but only a topic group. Usability of these words is probably much smaller than the content-type patterns. But the border between a content type and a topic group is rather thin.

Some words do not imply a specific content-type or a content topic, but still carry information about what can be expected on a page with this keyword in URL. Words like “about”, “product” or “forum” are widely used and understood. And thus are significant for a classification.

Patterns of automated systems

Another category of atomic components indicates that the URL was created and the content is maintained by an CMS, which certainly is a category of URLs. The most obvious is the word “wordpress”. About 70% of URLs containing word option were created by CMS Joomla. The pattern for Joomla is option=com_example found in query part of URL.

The joomla pattern leads to interesting discovery. The pattern does not necessarily have to be only one atomic component. It could be combination of two or more words.

With this in mind the query part of URL reveals another pattern. This time intentionally created by one company -

Urchin Tracker Module created for Google analytics tracking of a marketing campaign. The form is utm_. Input data set contain 513 of such links. This is enlarged to 1509, because the keyword is in one URL used multiple times as utm_source, utm_medium, utm_campaign, utm_term and utm_content. Of course this enlarges also counts for the related words in the bag as you can see in table.

Misleading and other words

Some words are misleading. Although they have high frequency, they're not a pattern. They're seasonally frequented words without long term significance. E.g. word “cup” with total count of 2212. Input data were collected at the time of football world cup.

Also words like “google” or “yahoo” have high frequencies, but no significance for a classification. In the paragraph about filtering the bag of word was already mentioned that adverbs and preposition words tend to have high frequencies, but cannot be filtered out.

CONCLUSION

This article clearly demonstrates that the Bag of words can be used with URL as input document. Also results show that there are relatively few word with high frequencies, which outlines their importance.

Resulting bag can be used for classification of URLs by several measures. Aside of this major result as minor result were identified two patterns which could be changed a used for constructing new URLs and revealing resources hidden in the deep web. Also were outlined several classes where the pattern is unknown.

Even through several problems with separation, stemization and filtering words, bag-of-words method was proven useful to be used against URL.

REFERENCES

[RHSZ] R. Hujer, *Patterns in URL* (Czech Technical University in Prague, 2011)

[RH10] R. Hujer, A. Kanai, *Numeric pattern in URL and its modification* (Groupware and Networking 78, 2010)

[CCLS] Li Cun-He, Lv Ke-qiang, *Hyperlink classification: A new approach to improve PageRank* (IEEE COMPUTER SOC, 2007)

[RFC3986] T. Berners-Lee, R. Fielding and L. Masinter, *RFC 3986 (STD66): "Uniform Resource Identifiers"* (2005)

[B40] D. Lewis, *Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval* (Proceedings of ECML-98, 1998)

[rfc1808] R. Fielding, UC Irvine: Relative Uniform Resource Locators, (http://tools.ietf.org/html/rfc1808.html, 1995)

[PrSt] K. Sparck Jones and P. Willet, *Readings in Information Retrieval*, (San Francisco: Morgan Kaufmann, ISBN 1-55860-454-4, 1997).

[PrStW] M.Porter: *The Porter Stemming Algorithm* (http://tartarus.org/martin/PorterStemmer/)