Automatic Web Page Classification

Jiří Materna

Natural Language Processing Centre
Faculty of Informatics, Masaryk University
Botanická 68a, 602 00, Brno, Czech Republic
xmaterna@fi.muni.cz
http://nlp.fi.muni.cz

Abstract. Aim of this paper is to describe a method of automatic web page classification to semantic domains and its evaluation. The classification method exploits machine learning algorithms and several morphological as well as semantical text processing tools. In contrast to general text document classification, in the web document classification there are often problems with short web pages. In this paper we proposed two approaches to eliminate the lack of information. In the first one we consider a wider context of a web page. That means we analyze web pages referenced from the investigated page. The second approach is based on sophisticated term clustering by their similar grammatical context. This is done using statistic corpora tool the Sketch Engine.

Key words: automatic classification, machine learning, web document, thesaurus

1 Introduction

1.1 Motivation

At the present time the World Wide Web is the largest repository of hypertext documents and is still rapidly growing up. The Web comprises billions of documents, authored by millions of diverse people and edited by no one in particular. When we are looking for some information on the Web, going through all documents is impossible so we have to use tools which provide us relevant information only. The widely used method is to search for information by fulltext search engines like Google[1] or Seznam[2]. These systems process list of keywords entered by users and look for the most relevant indexed web pages using several ranking methods. Another way of accessing web pages is through catalogs like Dmoz[3] or Seznam[4]. These catalogs consist of thousands web pages arranged by their semantic content. This classification is usually done manually or partly supported by computers. It is evident that building large catalogs requires a lot of human effort and fully automated classification

[1] http://www.google.com [2] http://search.seznam.cz [3] http://www.dmoz.org
[4] http://www.seznam.cz

Petr Sojka, Aleš Horák (Eds.): Proceedings of Recent Advances in Slavonic Natural Language Processing, RASLAN 2008, pp. 84–93, 2008.

systems are needed. However several systems for English written documents were developed (e.g. [1,2,3,4,5]) the approaches do not place emphasis on short documents nor on the Czech language.

1.2 Objective

Classical methods of text document classification are not appropriate for web document classification. Many of documents on the Web are to short or suffer from a lack of linguistic data. This work treats with this problem in two novel approaches:

- Experiments have proved that hypertext links in web documents usually direct to documents with similar semantic content. This observation leads to use these referenced web pages as an extension of the investigated one for the purposes of processing their linguistic data as well. However there are some restrictions. The referenced documents must be placed on the same server (to avoid joining advertisement or other non-related material) and a level of recursion must be limited. We experimentally set the limit to 2.
- The former method increases amount of linguistic data for the most part of documents enough but there is another problem. To use machine learning algorithms we need to build a high dimensional vector space where each dimension represents one word from or phrase. In spite of the fact that several machine learning algorithms are adjusted to high number of dimensions, in this case the high number of dimensions decreases algorithm accuracy and we have to proceed to dimensional clustering. The joining of two or more dimensions (in this case words) is based on using a special thesaurus built on training data. The method will be described more precisely in the Section *Term clustering*.

2 Preprocessing

In order to use machine learning algorithms we need to build a training data set. There were selected 11 domains (*Cestování, Erotika, Hry, Informační a inzertní servery, Kultura a umění, Lidé a společnost, Počítače a internet, Sport, Věda a technika, Volný čas a zábava, Zpravodajství*) according to the top-level domains in `http://odkazy.seznam.cz` catalog and for each domain collected 1 GB of sample data.

2.1 Data Cleaning

Despite of selecting restricted document content-types (HTML, XHTML) it is necessary to remove noise from the documents. An example of unwanted data is presence of JavaScript (or other scripting languages) as well as Cascading Style Sheets (CSS) and the most of meta tags. Elimination of such data was mostly done by removing *head* part of the document (except of content of

the *title* tag which can hold an important information about domain). As other unwanted data were marked all n-grams (n>10) where portion of non alphanumeric characters was grater than 50 %.

Very important issue of document preprocessing is charset encoding detection. However the charset is usually defined in the header of the document, it is not a rule. We have used a method of automatic charset detection based on byte distribution in the text [6]. This method works with a precision of about 99 %.

A lot of web sites allows user to chose language. Even some web pages on the Czech internet are primarily written in foreign language (typically in Slovak). With respect to used linguistic techniques, we are made to remove such documents from the corpus. The detection of foreign languages is similar to charset encoding detection based on typical 3-gram character distribution. There has been built a training set of Czech written documents and computed the typical distribution. Similarity of training data with the investigated documents is evaluated using cosine measure.

2.2 Corpus construction

Cleaned raw data serve as a groundwork for the training corpus construction. To represent corpus data we use vertical text with following attributes:

- **word** – original word form,
- **lemma** – the canonical form of a word. To get lemma we have used Ajka tagger [7] and disambiguator Desamb [8],
- **tag** – morphological tag of a word (obtained from Ajka).

To process data has been used corpus manager Manatee [9] which offer many statistical functions as well as the Sketch Engine tool [10]. This system can extract so called word sketches which provide information about usual grammatical context of terms in corpus and are used for the thesaurus construction.

3 Document Model

In order to use these data in machine learning algorithms we need to convert them into appropriate document models. The most common approach is vector document model where each dimension of vector represents one word (or token in corpus). There are several methods of representing the words.

Let m is number of documents in the training data set, $f_d(t)$ frequency of term t in document d for $d \in \{1, 2, \ldots, m\}$ and *Terms* set of terms $\{t_1, t_2, \ldots, t_n\}$.

3.1 Binary representation

Document d is represented as a vector $(v_1, v_2, \ldots, v_n) \in \{0, 1\}^n$, where

$$v_i = \begin{cases} 1 & \text{if } f_d(t_i) > 0 \\ 0 & \text{else} \end{cases}$$

3.2 Term frequency representation

Document d is represented as a vector $(v_1, v_2, \ldots, v_n) \in \mathbb{R}^n$, where

$$v_i = \frac{f_d(t_i)}{m}$$

3.3 Term Frequency – Inverse Document Frequency (TF-IDF)

Disadvantage of previous two methods may be a fact of treating with all terms in the same way – they are not weighted. This problem can be solved by using IDF coefficient which is defined for all $t_i \in Terms$ as:

$$IDF(t_i) = log_2 \left(\frac{m}{|\{j : f_j(t_i) > 0\}|} \right)$$

By combining TF and IDF we get:

$$v_i = \frac{f_d(t_i)}{m} \cdot log_2 \left(\frac{m}{|\{j : f_j(t_i) > 0\}|} \right)$$

For TF and TF-IDF methods is convenient to discretize their real values. The MDL algorithm [11] based on information entropy minimization has been used.

4 Term Clustering

The term clustering is based on a special dictionary. The dictionary is defined as a total function

$$s : Terms \rightarrow Rep$$

which assigns just one representative from $Rep \subseteq Terms$ to each member of $Terms$ set. The s function defines equivalence classes on $Terms$ by equivalence relation σ:

$$(a, b) \in \sigma \iff s(a) = s(b)$$

Reversely, let $C \in Terms/\sigma$, there always exists some function s. If r is an arbitrary member of C, then

$$s(x) = r \quad \text{for all} \quad x \in C$$

The construction of dictionary consits of following steps:

1. Finding characteristic set for each term $t \in Terms$.
2. Defining equivalence classes on $Terms$ set based on similarity of their characteristic set.
3. Dictionary function s definition.

4.1 Characteristic set

Characteristic set construction is mostly based on using the Sketch Engine and its word sketches. Word sketches are one-page automatic, corpus-based summaries of a word's grammatical and collocational behavior generated by Sketch Engine which takes as input a corpus of any language and a corresponding grammar patterns and which generates word sketches for the words of that language [10].

It suggest itself to look for similar word sketches and build a thesaurus. For each lemma l with sufficient frequency we get a list of similar words $SP_l = [w_1, w_2, \ldots, w_n]$ ordered by their indexes of similarity $i_1, \ldots, i_n$ with lemma l [12]. Lets define the *characteristic list* $CHL(l)$ for each lemma l from the corpus:

- if frequency of lemma l in the corpus is less than 100:
  $CHL(l) = [l]$
- else:
  $CHS(l) = [w_1, w_2, \ldots, w_k] : \forall i_j \in \{i_1, i_2, \ldots i_k\} : i_j \geq 0.1$

An example of characteristic list of lemma *auto* (car) is shown in Table 1.

Table 1. Characteristic list of lemma *auto*

auto	1
automobil	0.184
autobus	0.171
vůz	0.166
vozidlo	0.153
vlak	0.141
aut	0.133
tramvaj	0.126
lod'	0.124
letadlo	0.112
trolejbus	0.11

The table shows that the incorporated words are really semantically similar. However, there are some problems with homonyms and tagging errors (in this case term *aut*). The *characteristic set* is defined in the way of eliminating words occurred in the corpus more frequently in other senses than we currently treat with.

Let $CHL(l) = [w_1, w_2, \ldots, w_k]$ is the characteristic list of the lemma l, $S(l) = \{w_1, w_2 \ldots, w_k\}$ and $S_p(l) = \{w_i | i \leq k/p\}$ where $p \in \mathbb{R}^+$ is a constant coefficient. The *characteristic set* is defined as

$$CH(l) = \{w_i : q \cdot |S(w_i) \cap S_p(l)| \geq |S_p(l)|\}$$

where $q \in \mathbb{R}^+$ is an appropriate constant. The experiments have shown that the best values seem to be $p = 2, q = 2$.

4.2 Dictionary construction

When we have a characteristic set for each lemma from corpus it remains to define clustering and dictionary function s. Intuitively, the clusters are composed of terms with similar characteristic sets. In this work, the similarity is measured by Jaccard index, where similarity of terms a and b is defined as

$$j(a,b) = \frac{|CH(a) \cap CH(b)|}{|CH(a) \cup CH(b)|}$$

The clustering works on the principle of hierarchical clustering [13] using top-down method. Minimal similarity for joining sets was experimentally set to 0.45. These clusters define equivalence relation σ.

Let $freq(x)$ is a frequency of term x. We define dictionary function s: $\forall S \in Terms/\sigma, \forall a \in S : s(a) = b$ where $b \in S, freq(b) = max\{freq(x)|x \in S\}$. In the case of ambiguity the first possible lemma in lexicographical order is used.

Finally, when we have dictionary function s, we are able to replace all terms t in corpus by their representatives $s(t)$.

5 Attribute Selection

Even after application of the dictionary function there are a lot of different terms for using machine learning algorithms in the corpus and it is necessary to select the most convenient ones. Statistics provides some standard tools for testing if the class label and a single term are significantly correlated with each other. For simplicity, let us consider a binary representation of the model. Fix a term t and let

- $k_{i,0}$ = number of documents in class i not containing term t
- $k_{i,1}$ = number of documents in class $i$containing term t

This gives us a contingency matrix

$I_t \backslash$C	1	2	...	11
0	$k_{1,0}$	$k_{2,0}$	...	$k_{11,0}$
1	$k_{1,1}$	$k_{2,1}$	...	$k_{11,1}$

where C and I_t denote boolean random variable and $k_{l,m}$ denotes the number of observation where $C = l$ and $I_t = m$.

5.1 χ^2 test

This measure is a classical statistic approach. We would like to test if the random variables C and I_t are independent or not. The difference between observed and expected values is defined as:

$$\chi^2 = \sum_{l \in Class} \sum_{m \in \{0,1\}} \frac{(k_{l,m} - n \cdot P(C = l)P(I_t = m))^2}{n \cdot P(C = l)P(I_t = m)}$$

5.2 Mutual Information Score

This measure from information theory is especially useful when the multinominal document model is used and documents are of diverse length (as is usual). The mutual information score is defined as:

$$MI(I_t, C) = \sum_{l \in Class} \sum_{m \in \{0,1\}} \frac{k_{l,m}}{n} log \frac{k_{l,m}/n}{(k_{l,0} + k_{l,1}) \cdot (\sum_{i \in Class} k_{i,m})/n^2}$$

6 Classification and Evaluation

We have tested the classification using four algorithms (C4.5, *k*-nearest neighbors, Naîve Bayes classifier and Support machines) on 3,500 randomly chosen training samples and 1,500 testing examples. For testing has been used 10-fold cross validationi [14]. As an implementation, we have chosen open source data mining software Weka [15] for algorithm C4.5, *k*-nearest neighbors and Naîve Bayes classifier and LIBSVM [16] for Support Vector machines.

First, we compare preprocessing methods and selected machine learning algorithms on data without clustering and document extending. Next, the best-resulting method is chosen to test approaches presented in this paper. In Figure 1 you can see overall accuracy graphs of all presented algorithms and methods of document model representation. The best results with 79.04 % of overall accuracy have been acquired using Support vector machines algorithm, term frequency document model and MI-score selection of attributes.

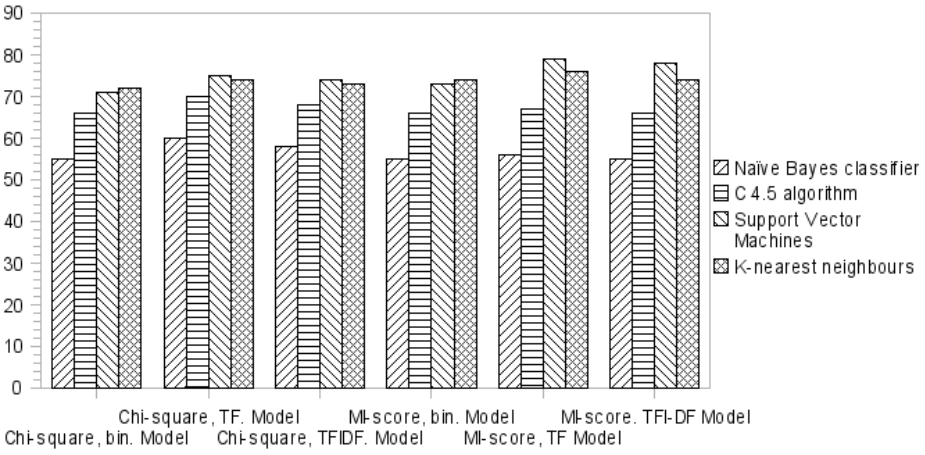

Fig. 1. Preprocessing and classification algorithms

Figure 2 shows dependency of overall accuracy on attribute number without clustering, with clustering based on same lemmas and with clustering

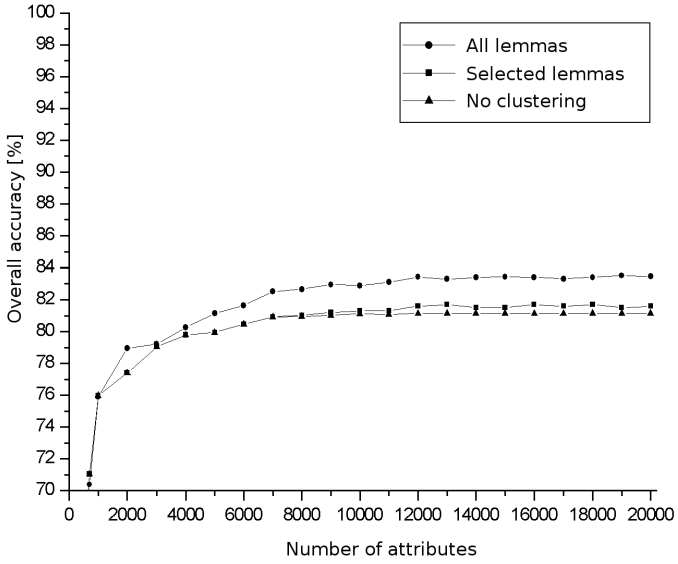

Fig. 2. Clustering methods

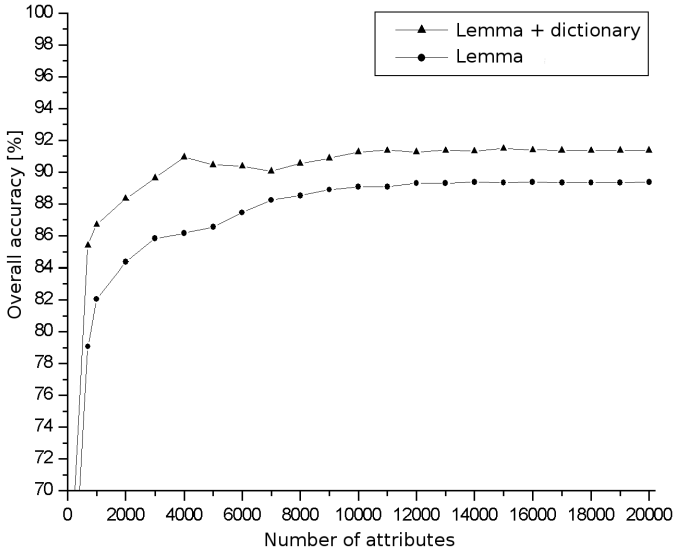

Fig. 3. Extending by referenced documents

based on selected lemmas. In the third case, only nouns, adjectives, verbs and adverbs have been selected. You can see that overall accuracy in all cases grows

till about 12,000 attributes. After this threshold the overall accuracy does not vary significantly. The best result (83.4 %) was acquired using clustering based on same lemmas.

Finally, Figure 3 shows result of experiments with extended documents, clustering based on same lemmas and on both lemmas and dictionary. The overall accuracy growth from previous experiment is about 5.9 % for lemma based clustering and 8.2 % for dictionary based clustering.

7 Conclusion

We have presented a method of automatic web page classification into given 11 semantic classes. Special attention has been laid on treating with short documents which often occur on the internet. There have been introduced two approaches which enable classification with overall accuracy about 91 %. Several machine learning algorithms and preprocessing methods have been tested. The best result has been acquired using Support vector machines with linear kernel function (followed by method of k-nearest neighbors) and term frequency document model with attribute selection by mutual information score.

Acknowledgments. This work has been partly supported by the Academy of Sciences of Czech Republic under the projects 1ET100300419 and 1ET200610406, by the Ministry of Education of CR within the Center of basic research LC536 and in the National Research Programme II project 2C06009 and by the Czech Science Foundation under the project 407/07/0679.

References

1. Asirvatham, A.P., Ravi, K.K.: Web page categorization based on document structure (2008) `http://citeseer.ist.psu.edu/710946.html`.
2. Santini, M.: Some issues in automatic genre classification of web pages. In: JADT 2006 – 8èmes Journèes internationales d'analyse statistiques des donnés textuelles, University of Brighton (2006).
3. Mladenic, D.: Turning Yahoo to automatic web-page classifier. In: European Conference on Artificial Intelligence. (1998) 473–474.
4. Pierre, J.M.: On automated classification of web sites. **6** (2001) `http://www.ep.liu.se/ea/cis/2001/000/`.
5. Tsukada, M., Washio, T., Motoda, H.: Automatic web-page classification by using machine learning methods. In: Web intelligence: research and development, Maebashi City, JAPON (23/10/2001) (2001).
6. Li, S., Momoi, K.: A composite approach to language/encoding detection. 9th International Unicode Conference (San Jose, California, 2001).
7. Sedláček, R.: Morphemic Analyser for Czech. Ph.D. thesis, Faculty of Informatics, Masaryk University, Brno (2005).
8. Šmerk, P.: Towards Morphological Disambiguation of Czech. Ph.D. thesis proposal, Faculty of Informatics, Masaryk University, Brno (2007).

9. Rychlý, P.: Korpusové manažery a jejich efektivní implementace (in Czech). Ph.D. thesis, Faculty of Informatics, Masaryk University, Brno (2000).
10. Kilgarriff, A., Rychlý, P., Smrž, P., Tugwell, D.: The Sketch engine in practical lexicography: A reader. (2008) 297–306.
11. Fayyad, U.M., Irani, K.B.: On the handling of continuous-valued attributes in decision tree generation. Machine Learning **8** (1992) 87–102.
12. Kilgarriff, A.: Thesauruses for natural language processing. Proc NLP-KE (2003).
13. Berka, P.: Dobývání znalostí z databází. Academia (2003).
14. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: IJCAI. (1995) 1137–1145.
15. Witten, I.H., Frank, E.: Data mining: Practical machine learning tools and techniques. Technical report, Morgan Kaufmann, San Francisco (2005).
16. Chang, C.C., Lin, C.J.: LIBSVM: a Library for Support Vector Machinesi. Technical report, Department of Computer Science National Taiwan University, Taipei 106, Taiwan (2007).