Volume : 4 | Issue : 1 | January 2015 • ISSN No 2277 - 8179

Research Paper

Automatic Web Page Classification Based on Firefly and Linear Algorithm

Computer Science

KEYWORDS : Classification, Web Page Classification, Optimization, Feature Selection

Yogita P. Nemade	M.Tech Scholor, Computer Science Department, Patel College of Science and Technology (PCST), Indore (M.P.), India,
Praveen Bhanodia	Assi. Professor and Head, Computer Science Department, Patel College of Science and Technology (PCST), Indore (M.P.), India
Diamond Jonawal	Assi. Professor, Computer Science Department, Patel College of Science and Technology (PCST), Indore (M.P.), India

ABSTRACT

Aim of this paper is to describe a method of automatic webpage classification. The classification method exploits Firefly algorithms and linear classifier as well as semantically 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.

1 INTRODUCTION

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[1,2] .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 full text search engines like Google1 or Seznam2. 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 Dmoz3 or Seznam4 or weka dataset . These catalogs consist of thousands web pages arranged by their semantic content. This classification is usually done manually or partly supported by computers[3]. It is evident that building large catalogs requires a lot of human effort and fully automated classification systems are

needed. However several systems for English written documents were developed [4,5] the approaches do not place emphasis on short documents nor on the Czech language.

2 PREPROCESSING

In order to use FA & Linear algorithms we need to build a training data set. 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 greater 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 char set 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 choose language. Even some web page son he Czech internet is 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.

3 CONSTRUCTION

Cleaned raw data serve as a ground work 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.
– tag – morphological tag of a word

To process data has been used corpus manager Manatee [9] which offers 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.

A. TEXT SOURCES FOR WEB PAGE REPRESENTATION

Web pages can be represented in various ways. Maybe the simplest way to represent a web page is to extract the text found within the BODY element. This representation does not exploit the peculiarities of web pages, i.e. HTML structure and the hyper textual nature of web pages.

B. HTML structure

By exploiting HTML structure [4] for web page representation we can choose how a term is representative of the page considering the HTML element it is present in. For example, we can represent a web page using only the words of the title, that is to say the words extracted from the TITLE element. For obtaining good performance in web page representation exploiting HTML structure is important to know where the more representative words can be found. For example, we can think that a word present in the TITLE element is generally more representative of the document's content than a word present in the BODY element. We tested five different text sources for web page representation,[8] namely:

BODY, the content of the BODY tag;
META, the meta-description of the META tag;

TITLE, the page's title;

Volume : 4 | Issue : 1 | January 2015 • ISSN No 2277 - 8179

MT, the union of META and TITLE content;

4. THE GENERAL PROBLEM OF WEB PAGE CLASSIFICATION

The general problem of the web page classification can be divided into multiple sub-problems such as subject classification, functional classification and other types of Classification.

1. Subject Classification:- Subject classification is concerned about the subject or topic of the web page. For e.g., categories of online newspapers like finance, sport, technology, are instances of subject classification.

2. Functional classification:-Functional classification is concerned with function or type of a Web page. For e.g.,determining a page is a "personal homepage" or a "coursepage" is an instance of a functional classification.

3. Based on the number of classes in the problem, the classification can be divided into binary classification and multiclass classification. Binary Classification:- In binary classification there is only one class label. The Classifier looks an instance and assigns it into the specific class or not. Here instances of the specific class are called as relevant instances, and the others are named as non-relevant instances. Multiclass Classification:-If there are more than one class, this is called as multiclass classification. Classifier also assigns an instance to one of the multiple classes.

4. Based on the number of classes that can be assigned to an instance, the classification can be divided into single-label classification and multilabel classification:-Single-label Classification:-In single-label classification, only one class label is to be assigned to every instance, while in multilabel classification; more than one class can be assigned to an instance. When a problem is multi-class, e.g. four-class classification, it means four classes are involved Arts, Business, Sports and Computers. It can either be single-label, exactly where one class label can be assigned to an instance. Multi-label Classification:- multi-label, where an instance can related to any one, two or all of the classes.

5. Based on the type of class assignment, the classification can be divided into hard classification and soft classification. Hard Classification :-In hard classification , an instance can either be or not be in a particular type of class, without an intermediate state; while in soft classification , an instance can be predicted to be in some class with some likelihood(often a probability distribution across all classes).

6. Based on the organization of categories, the Web page classification can also be divided into flat classification and hierarchical classification. In the flat classification, categories are considered as parallel, i.e., one category does not supersede another. But in hierarchical classification categories are organized in a hierarchical tree-like structure, in which each category may have a number of subcategories [7].

5. METHODOLOGY

Here we use Firefly algorithm & Linear Algorithm to find the strongest feature of web dataset. Figure 1 shows the Classification of Web Dataset.

Firefly algorithm is a based wrapper technique which finds the best features for Web pages, to make fast and accurate classification. Firefly Algorithm (FA) is a recent search and optimization technique, which was first introduced by Xin-She Yang in 2008[10]. The primary purpose for a firefly's flash is to act as a signal system to attract other fireflies. The algorithm constitutes a population-based iterative procedure with numerous agents (perceived as fire flies)

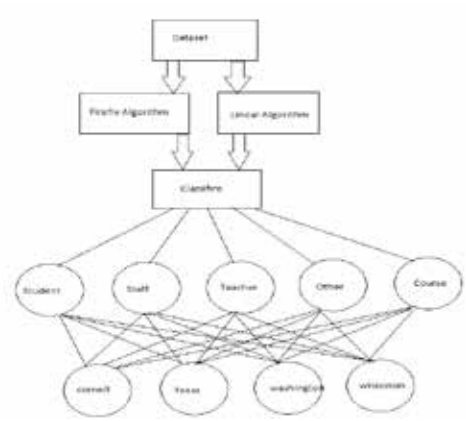

Figure 1: Classification of Web Dataset

concurrently solving a considered optimization problem. Agents communicate with each other via bioluminescent glowing which enables them to explore cost function space more effectively than in standard distributed random search. Intelligence optimization technique is based on the assumption that solution of an optimization problem can be perceived as agent (firefly) which glows proportionally to its quality in a considered problem settings.

Firefly Algorithm

Objective function $f(x)$, $x = (x_1, \ldots, x_d)^T$

Generate initial population of fireflies x_i $(I = 1, 2, \ldots, n)$

Light intensity I_i *at* x_i *is determined by* $f(x_i)$

Define light absorption coefficeint γ

while $t <$ *MaxGeneration* **do**

for i = 1 : n all n fireflies **do**

for j = 1 : i all n fireflies **do**

if $I_j > I_i$ **then**

Move firefly i towards j in d-dimension;

end if

Attractiveness varies with distance r via exp[$-\gamma r$]

Evaluate new solutions and update light intensity

end for

end for

Rank the fireflies and find the current best

end while

Postprocess results and visualization

Figure 2: Firefly Algorithm

Consequently each brighter firefly attracts its partners (regardless of their sex), that makes the search space being explored more efficiently. The main rules of the algorithm are as follows:

• All fireflies are unisex and they will move towards more attractive and brighter ones regardless of their sex.

• The degree of attractiveness of a firefly is proportional to its brightness, Also the brightness may decrease as the distance from the other fire flies increases due to the fact that the air absorbs light. If there is not a brighter or more attractive firefly than a particular one it will then move randomly.

• The brightness or light intensity of a fire fly is determined by the value of the objective function of a given problem.

Linear Algorithm

Input:

- Positive training examples, POS
- Unlabeled examples (sample of universal set), U

Output:

- a SVM

Algorithm:

- 1-DNF := construct_1-DNF (POS, U); // ... (1)
  M_1(neg), S_1(pos) := 1-DNF.classify (U); // ... (2)
  NEG: = ϕ; I := 1;
  do {
  - NEG:= NEG$\cup$ M_i(neg);
  SVM:= construct_SVM (POS, NEG); // ... (3)
  M_{i+1}(neg), S_{i+1}(pos) := SVM.classify (S_i(pos));
  // ... (4)
  I := I + 1;
  } while (M_i(neg) $\neq$ ϕ);
  return SVM;

Figure 3: Linear Algorithm

Rationale:

1-DNF learns from POS and U ... (1), and extracts the strongest negative (M1(neg)) from U ... (2). (The remainder is S1(pos).) Save the M1(neg) into NEG, and construct a SVM from POS and the NEG ... (3). The SVM classifies S1(pos) into the secondly strong negative (M2(neg)) and the remainder (S2(pos)) ... (4). Accumulate the M2(neg) into NEG, and construct a SVM again from POS and NEG ... (3). The SVM classifies S2(pos)into M3(neg) and S3(pos) ... (4). We iterate these processes until Mi(neg) becomes empty set.

RESULTS:

All the implementation for the experiments were made in Java programming language under Eclipse environment [11]. The proposed method was tested under Microsoft Windows 7 operating system. The hardware used in the experiment had 4 GB of RAM and intel core i5 processor.

In this study the firefly and linear algorithm selects a predefined number of features . Experimental results of proposed FA based method finds the strongest features and use this for classification.

Figure 4 shows results as the popup window clearly shows that the required person is student and from Cornell university and provides details of that student using proposed automatic web page classification method from 4 different universities data of WebKB dataset.

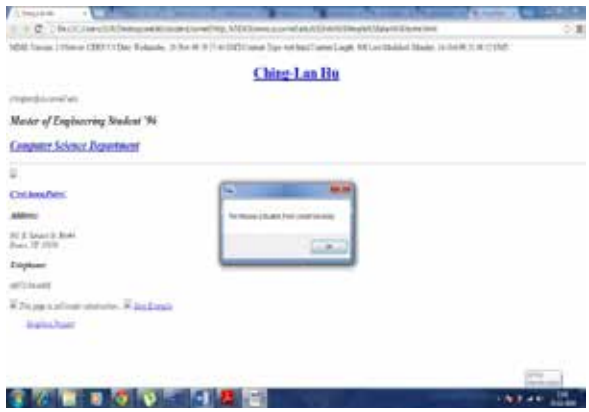

Figure 4: Classification from WebKB dataset

Similarly figure 5 shows results for next required person which is student from Wisconsin university with its details by automatic web page classification.

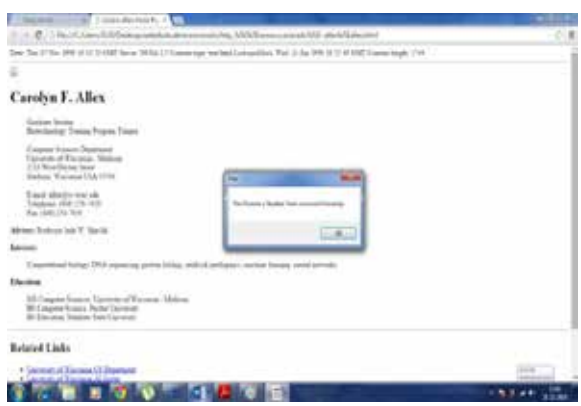

Figure 5: Classification from WebKB dataset

CONCLUSIONS:

In this paper, a preliminary empirical evaluation of the performance of a Firefly Algorithm plus Linear algorithm. Using this method we find the strongest feature from both classifier & use this for classification. This optimal method is a representation of a set of correlated users which appears to have similar web usage behavior under subset of pages of a web site. Hence it can be aggregated to represent the optimal usage profile (i.e., a collection of web pages or URLs) for a web site. Usage profile discovered is effective in capturing user-to-page relationship and similarities at the level of user sessions. This knowledge can be used in the applications like personalization, target marketing etc.

REFERENCE

[1] Cheng, Y., Church, G.M.: Biclustering of Expression Data. In: International Conference on Intelligent Systems for Molecular Biology, 2000, pp. 93–103. | [2] Coelho, G.P., de França, F.O., Von Zuben, F.J.: Multi-Objective Biclustering: When Non dominated Solutions are not Enough. J. Math. Model Algor. Vol. 8, 2009, pp:175–202. | [3] Das C, MajiP ,Chattopadhyay S, A Novel Biclustering Algorithm for Discovering Value-Coherent Overlapping σ-Biclusters, Advanced Computing and Communications, 2008, pp:148-156,. | [4] David Beasley, David R. Bull, and Ralph R. Martin, An overview of genetic algorithms: Part 2, research topics. University Computing, Vol.15, No. 4,1993, pp: 170–181. | [5] de Castro, P.A.D., de França, F.O., Ferreira, H.M., Von Zuben, F.J., Applying Biclustering to Perform Collaborative Filtering. In: International Conference on Intelligent System Design and Applications, 2007, pp. 421–426. | [6] de Castro, P.A.D., de França, F.O., Ferreira, H.M., Von Zuben, F.J. ,Applying Biclustering to Text Mining: An Immune-Inspired Approach.In: de Castro, L.N., Von Zuben, F.J., Knidel, H. (eds.) ICARIS 2007.LNCS, vol. 4628, pp. 83–94. Springer, Heidelberg, 2007. | [7] de França, F.O., Coelho, G.P., Von Zuben, F.J.,bicACO: An Ant Colony Inspired Biclustering Algorithms. Ant Colony Optimization and Swarm Intelligence, Lecture Notes in Computer Science Volume 5217, 2008, pp 401-402. | | [8] Divina, F., Aguilar-Ruiz, J.S.: Biclustering of Expression Data with Evolutionary Computation. IEEE Trans. Knowl. Data Eng. Vol. 18, 2006, pp: 590–602. | [9] Kennedy, J. and Eberhart, R.C., A discrete binary version of the particle swarm algorithm, Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation, IEEE International Conference, vol.5,pp.:4104 - 4108 ,1997. | [10] Madeira, S.C., Oliveira, A.L.: Biclustering Algorithms for Biological Data Analysis: A Survey. IEEE/ACM Trans. Comput. Biol. Bioinform. Vol.1,, 2004, pp:24–45. | [11] Eclipse environment available at the website of oracle http://www.oracle.com/techwork/developer-tools/eclipse /downloads/index.html. |