International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 16 (2017) pp. 5660-5664

A Machine Learning approach to Classify web documents of freelancing and remote work in IT field

Ms. Sharmila Shinde
Ph.D. research scholar,
Bharati Vidyapeeth Deemed University,
College of Engineering,
Pune, India.

Orcid Id: 0000-0003-0931-093X

Dr. Prasanna Joeg
Computer Engineering Department,
Bharati Vidyapeeth University,
Pune, India.

Orcid Id: 0000-0002-7 625-4205

Dr. Sandeep Vanjale
Computer Engineering Department,
Bharati Vidyapeeth University,
Pune, India.

Orcid Id: 0000-0001-5944-7120

Abstract

Vertical Search Engine (VSE) indexes only those pages which are relevant to the predefined subject. It is necessary to determine which kind of information in a web page is relevant. This paper addresses the issue of relevancy of a web page which has the content of freelance and remote work postings. For this, we propose a machine learning approach which checks the content of the web page. It extracts keyword occurrences within a web page. It then looks for relevant keywords and classifies the web page as positive if it contains remote work/freelance job postings. This approach incorporates incremental training to improve accuracy. Three experiments were designed by changing the number of relevant web pages given for training. Performance is measured using accuracy, precision, and recall. Our results suggest that only a few relevant web pages need to be given for training to get a high accuracy.

Keywords: Vertical search engine, Document Classifications, Training set, Machine Learning.

INTRODUCTION

To make an efficient search in a specific domain many Vertical Search Engines (VSE), have been built. Though VSE is a good alternative for a specific search for targeted user, it is very difficult to design a vertical search engine. A major issue related to VSE is to locate seed URL or relevant pages of a specific domain. There are many approaches such as content and link analysis which filters irrelevant documents, still, none of them is very promising to provide accurate results [1].

The categorization of a web page into predefined category needs human expertise which is a time consuming and costly. This process can be automated using machine learning approach. Document classification algorithms play an important role in filtering relevant pages. In literature, there are three techniques of classification, supervised, unsupervised and semi-supervised. In last few years there has been a lot of work focused on automatic document classification which involves accurate model construction to categorize each class based on these features. Many algorithms like the neural network, decision tree. Support vector machine, naïve-Bayesian, natural language processing is widely used for document classification [2].

Over the last few years, the freelancing domain saw a 26 percent increase in the number of jobs posted on its site, while 50% of the workforce working in IT related fields currently has jobs that can be done remotely, according to Global Analytics [3] [4]. It is now very important for job seekers to be able to find freelance/remote job postings easily. This work focuses on finding specific web pages that contains freelance and remote work postings across the all geography. It is essential to locate a set of relevant web pages which are given to the classification algorithm as positive web pages. The relevancy of web pages is checked with the help of content present inside it. Some other web pages that contain full-time job postings and no freelancing work are given as negative web page to the classification algorithm. To build a training set we extracted frequently occurred keywords. These keywords are provided to web page classifier for training. The classifier will generate a set of positive keywords (relevant to remote job postings) and negative keywords (not relevant to any remote job postings). Once the training phase is done, performance is checked on the testing set.

In the first step in classification, we manually labeled web pages based on content available on that page. e.g. 'I need a freelance java developer', 'we are looking for remote PHP developer', and 'need work from home website developer' are the positive web page. All other web pages are labeled as the negative web page. This labeled dataset after preprocessing is used for testing and training. Once web page classifier is trained, it classifies the given set of web page into positive web page and negative web page.

This paper is organized into five sections. Section 2 discusses related work in the vertical search engine and various techniques related to web page classification. Section 3 discusses our approach; Section 4 gives an idea about experimental setup and performance evaluation.

RELATED WORK

To improve the efficiency of VSE it is essential to filter noisy pages. To filter irrelevant pages there are many techniques in the research literature. Relevance can be decided with the help of occurrence of particular keyword set.

TF-IDF is measured and the similarity score is compared with the set of the relevant documents based on some threshold [5]. There are many machine learning approaches for text classification. Naive Bayesian is widely used to calculate the probability which predicts the document that belongs to each category [6].

Neural network method learns patterns by adjusting weights among nodes based on learning examples. The network is trained with the help of learning example to judge the category of a document [7]. Another very popular approach used for text classification is support vector machine. It finds a hyper plane that separates two classes [8] [9].

Many machine learning algorithms play important role in semantic web. It is used to create ontologies. Association rule and clustering algorithms are used to extract knowledge from web pages and further used to build ontologies [10].

Web mining research is mainly categorized as web content, web usage, and web structure mining [11]. There are many research papers on document categorization and clustering. Web structure mining mostly analyzes in links and out links which contribute in ranking results of search engine. Web usage mining analyzes weblogs to find usage pattern of the user. This research is very useful to recommend and build a user profile. Web pages are very diverse due to its structure, length, language, and formats. It is also dynamic in nature. All these issues make it very difficult for search engine to create an index for the web. As the web has hyperlink structure so many researchers have suggested approaches to find relevancy of topic based on anchor text or adjacent text to predict the content of target page [12].

To guide a search process, context focused Crawler uses a Naïve Bayesian classifier [13]. Various characteristics of Web mining have also discussed the use of link structure to improve Web classification [14]. Wang and Hu suggested pattern learning table layout using decision tree and SVM [15].

OUR APPROACH

A. Data Preparation

As we are using the supervised approach we need a collection of URLs. which can be used as training and testing set. The web pages at the URLs are mix web pages. Some of these web pages have postings for freelance/remote jobs. A web page is considered positive if it has freelance/remote job posting. Otherwise, the web page is considered to be negative.

We downloaded a certain n number of documents from different job websites. The number n will be explained below. We manually labeled each of these documents as positive and negative.

The content from each web page is extracted and pre-processed to eliminate special characters, stop words and converts the uppercase texts to lowercase. After preprocessing, nouns are extracted as keywords

B. Web Page Classification

Our classification algorithm is as follows: It checks the number of times each keyword has occurred. Each keyword in a positive web page is considered positive along with the frequency of occurrence (count) and stored in a positive set (PK). Similarly, negative keywords are also stored from the negative set (NK). If the count is below a set threshold then the keyword is not added to the keyword set, either positive or negative. If the keyword is common in both sets then the occurrence with higher count is retained. This keyword is deleted from the other set. In this manner, we process all n web pages. The algorithm is given in fig. 1. The output of this classification algorithm is a set of positive keywords and negative keywords.

Input: Preprocessed document P

I. For each document P:
   1. Keywords (Nouns) are extracted from the preprocessed text and stored in a separate positive and negative text file with its occurrence count.
   2. If a keyword is present in both text files, frequency count of the keyword is the subtraction of negative count from the positive count.

      score = positiveCount - negativeCount;
      if(score>0)

      - Add keyword to set PK

   else

      - score = Math.abs (score)

      if(score>0)

      - Add keyword to set NK.

II. Output: Set of positive and negative keywords, PK and NK

Figure 1: Classification Algorithms

C. Testing

The We again downloaded a larger set of m web pages(documents). Our testing algorithm is as follows: These web pages are preprocessed as before and a new set of keywords is extracted. The number of times keywords from PK set and

NK set occur is calculated. If the keywords from PK set have a higher total count than the total count of keywords of NK set, then this new web page is classified as positive. Otherwise, web page is classified as negative. The testing algorithm is given in fig. 2

1. Input: Set of M document, Positive set PK and Negative set NK.
2. Preprocess each document and extract keywords.
3. Compare count of extracted keywords with PK and NK.
4. If it maps with more positive keywords than negative keywords it classifies it as positive else negative document.
5. Output: Set of classified document.

Figure 2: Testing Algorithm

EXPERIMENTAL SETUP AND PERFORMANCE EVALUATION

We have performed three separate experiments and checked the performance of our classification algorithm. To generate web pages a sample set of initial URLs was used as given in Table 1. The web pages at these URLs will form the initial set of n web pages for training.

TABLE 1: SAMPLE INITIAL SET OF URLS:

Website: https://www.freelancer.com	
Positive URL	***Negative URL***
- https://www.freelancer.com/projects/Website-Design/complete-Gallery-Website-post-photos/ - https://www.freelancer.com/projects/NET/Net-MVC-Developer-12751646/ - https://www.freelancer.com/projects/Java/PROGRAMMERS-WHO-HAVE-READY-CODE/ - https://www.freelancer.com/projects/Website-Design/Build-Fashion-Commerce-Platform-site-12752504/ - https://www.freelancer.com/projects/Software-Architecture/Write-some-software-12751112/	- http://enable-javascript.com - https://www.freelancer.com/showcase - https://www.freelancer.com/contest/browse/1/ - https://jobs.github.com/positions/80fb4642-5067-11e7-8561-275d670e71e6 - http://jobsflare.com/?action=ViewJobDetail&id=4525

We analyzed the performance of each experiment using precision, recall, and accuracy. Precision measures the fraction of the documents correctly classified as relevant, while recall measures the fraction of relevant documents retrieved from the data set. Accuracy measures the prediction correctness of the classifiers. These measures are used in classification evaluation and have been implemented as follows:

Precision=True Positive(TP)/Predicted

Recall=True Positive(TP)/Actual positive

Accuracy=((True Positive(TP)+True Negative(TN))/Total (n))

Experiment 1

Initially, we selected n=10 and extracted ten web pages. Five of these web pages had freelance/remote job posting content. These web pages are given as positive pages to the classification algorithm. Out of the other five web pages, two had full-time job postings and no freelance/remote job content. The remaining three are general informative web pages with no remote job posting content. These web pages were given as negative pages to the classification algorithm. The algorithm generated a set of positive (PK1) and negative (NK1) keywords separately which were used as a training set for this experiment.

To check the performance, a set of 400 web pages, half of which contained freelance/remote job posting content was downloaded. Other half contained full-time and general informative content. These are given as an input to testing algorithm. Based on keywords in PK1 and NK1 it will classify these web pages into positive and negative classes.

Experimental results show that it classified negative web pages more accurately than positive web pages. The set PK1 is not sufficient to generate some positive keywords and hence some positive pages are not classified accurately. However, even on this smaller set of training data accuracy is 76%, precision 78.2% and recall 72%.

TABLE 2. PERFORMANCE FOR N=10

	Predicted Positive	**Predicted Negative**	**Total**
Actual Positive	144 (TP**)**	56 (**FN)**	200
Actual Negative	40 (**FP)**	160 (**TN)**	200
Total	184	216	400

Experiment 2

For this experiment, n is now increased to 100 in an effort to improve the accuracy. We have extracted 100 web pages including previous ten. Again we chose fifty of them with freelance/remote job content and given as positive web pages to the classification algorithm. Other fifty have full-time job posting and general information content but no freelance/remote job content and given as negative web pages to the classification algorithm. To build another set of positive keyword (PK2) and negative keywords (NK2) we ran classification algorithm. It is observed that more new keywords are added. The positive keyword set PK2 is increased by 60% over PK1 and negative keyword set NK2 is increased by 45 % than NK1. Also, few keywords which were common in both sets are swapped from positive class to negative class and vice versa which is shown in table 3.e.g.'App' keyword from PK1 is shifted to NK2.

Performance is evaluated on the same set of 400 web pages which is taken for the first experiment. It is observed that prediction accuracy is 82% and it improved for positive class as training set become more accurate due to the more appropriate addition of keywords in respective sets. The precision and recall is also 82%.

For the first two experiments we used static training set.

TABLE 3. PERFORMANCE FOR N=100

	Predicted Positive	**Predicted Negative**	**Total**
Actual Positive	164	36	200
Actual Negative	36	164	200
Total	200	200	400

Experiment 3

In the third experiment we have increased n=1000. For this, we have extracted 1000 web pages (500 positive and 500 negative). In this 500 web pages had freelance/remote job content and these pages are given as positive pages to the classification algorithm. Other 500 web pages had full time and general information content which are given as negative pages to the classification algorithm. We have included previous 100 web pages in this list. The training set is built with incremental learning approach by using a set of 1000 web pages. Each web page is preprocessed and provided to the classification algorithm which generates new updated set of positive keyword (PK3) and negative keyword (NK3). These classified web pages are again preprocessed and training set is updated after every new input web page. In this experiment we again observed keywords switching from PK2 to NK3 and NK2 to PK3.The positive set PK3 is larger than PK2 by 45% and negative keyword set NK3 is larger than NK2 by 25%. 'Project' and 'team' keywords are switched from NK2 to PK3.

Incremental updating of keyword sets helps to prepare more accurate training set, which is resulted in better accuracy.

In third experiment prediction accuracy for the positive class is more than negative class. As more web pages are given for training, positive keyword set become more accurate which also helps to improve overall accuracy. The result shows the accuracy is 86.37%, precision 85.32% and recall 87.87%

TABLE 4. PERFORMANCE FOR N=1000

	Predicted Positive	**Predicted Negative**	**Total**
Actual Positive	442	58	500
Actual Negative	74	426	500
Total	516	484	1000

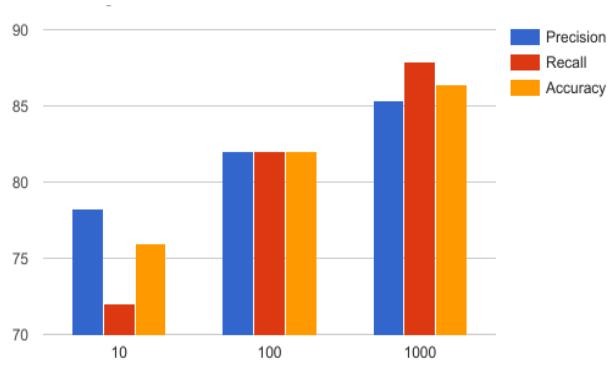

Figure 3: Comparative analysis of precision, recall and accuracy for different n

For further analysis, we extracted keywords from 100 full-time job postings and compared with freelancing and remote work keyword set. It was found that 30% keywords were common in both sets. We also gave this set of full-time job posting web pages for the classification algorithm. This algorithm classified 70 Percent full-time web pages as negative. It shows that web page classification algorithm works well to detect freelancing and remote job content pages.

CONCLUSION AND FUTURE SCOPE

As the number of freelancing opportunities grows this work will help a large number of people find suitable opportunities. In this paper, we discussed machine learning approach to classify the web document into positive and negative document and found the experimental results to be encouraging. From the experiments on the real domain-specific web pages, the proposed classification method shows better performance. As the training set get more evolved, the accuracy is also improved. We believe that the proposed approach is also useful for various Web applications, especially for vertical search engine development to locate relevant seed URLs (web pages).

REFERENCES

[1] Chau, M., & Chen, H. (2008). A machine learning approach to web page filtering using content and structure analysis. Decision Support Systems, 44(2), 482-494. DOI:10.1016/j.dss.2007.06.002

[2] W. A. AWAD, "Machine learning algorithms in web page classification," International Journal of Computer Science & Information Technology (IJCSIT) Vol 4, No 5, October 2012 DOI : 10.5121/ijcsit.2012.4508

[3] https://www.freshbooks.com/blog/freelance-jobs

[4] http://www.businessinsider.in/The-Top-25-Companies-For-Work-From-Home-Jobs/artcle show/38812987.cms

[5] O. Baujard, V. Baujard, S. Aurel, C. Boyer, R.D. Appel, Trends in medical information retrieval on the Internet, Computers in Biology and Medicine 28 (1998) 589–601.

[6] K.M.A.Chai,H.L.Chieu,H.T.Ng,Bayesian online classifiers for text classification and filtering, Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Tampere, Finland, Aug 2002, pp. 97–104.

[7] A. Selamat, S. Omatu, Web Page Feature Selection and Classification Using Neural Networks, Information Sciences [J], 2004, vol.158, PP:69-88

[8] Rung-Ching Chen,Chung-Hsun Hsieh Web page classification based on a support vector machine using a weighted vote schema, Volume 31, Issue 2, August 2006, Pages 427–435

[9] Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee, and Khairullah khan, "A Review of Machine Learning Algorithms for Text-Documents Classification," Journal of Advances in Information Technology, Vol. 1, No. 1, pp. 4-20, February, 2010.doi:10.4304/jait.1.1.4-20

[10] Rihab Idoudi, Karim Saheb Ettabaa, Basel Solaiman,Kamel Hamrouni, " Ontology Knowledge mining based Association Rules Ranking ", 20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2016, 5-7 September 2016, York, United Kingdom

[11] Walid Ahmed Fouad A Comparative Study Of Web Document Classification Approaches Proceedings of the 37th International Conference on Computers and Industrial Engineering, October 20-23, 2007.

[12] Chau, M., & Chen, H. (2008). A machine learning approach to web page filtering using content and structure analysis. Decision Support Systems, 44(2), 482-494. DOI:10.1016/j.dss.2007.06.002

[13] M. Diligenti, F. Coetzee, S. Lawrence, C.L. Giles, M. Gori, Focused crawling using context graphs, Proceedings of the 26th International Conference on Very Large Databases, VLDB 2000, Cairo, Egypt, 2000, pp. 527–534.

[14] Fürnkranz, J.: Web mining. In: Maimon, O., Rokach, L., (eds.): Data Mining and Knowledge Discovery Handbook. Springer-Verlag, Berlin (2005) 899-920

[15] Kozanidis L. (2008) An Ontology-Based Focused Crawler. In: Kapetanios E., Sugumaran V., Spiliopoulou M. (eds) Natural Language and Information Systems. NLDB 2008. Lecture Notes in Computer Science, vol 5039. Springer, Berlin, Heidelberg