A Neural Network Language Document Representation Technique for Web-Page Classification

Osanyin Quadri A.
Department of Computer Science
The Federal Polytechnic Ilaro,
Ogun State, Nigeria

Ajose-Ismail B. M.
Department of Computer Science
The Federal Polytechnic Ilaro,
Ogun State, Nigeria

ABSTRACT
The task of assigning a web page to the correct category is getting cumbersome because of the influx of digital documents on the World Wide Web. The performance of applications such as web directories, question and answering system, web content filtering systems depends on the key performance of automatic web page classification systems. From extant literature, the performance of web page classification system depends on adequate textual representation of the web content. Several statistical document representation techniques such as bag of words models, n-grams models and topic models have been proposed by authors to capture the real semantics of web documents but are fraught with several challenges such as semantic mismatch, multiple meanings of words. Thus, this paper proposes a recent neural network language model (Doc2Vec) which utilizes document embedding's to solve the document representation problem of web page classification system. Results obtained confirms the earlier assumption that Doc2Vec performs robustly on very high dimensional text such as web documents, it also capture the real semantics of the web document.

Keywords
Classification, Document embedding's, Machine learning, Document representation, Web Page classification, Doc2Vec

1. INTRODUCTION
According to google index, the volume of digital documents available online is over 130 trillion pages and its growing exponentially as a result of increased usage of the internet. Finding relevant and timely information from these documents are important for many applications such as web directories provided by different search engines like Google and Yahoo, relevant search results and question answer system [1], [2]. Also, with the influx of data from smartphones, social media websites and several data driven applications, free flowing text are on the rise [3]. Automated text categorization is the key technology for this tasks. Web page classification, also known as web page categorization, is the process of assigning a web page to one or more predefined category or labels (Qi & Davison, 2009). In formal terms, any web classification system is designed to accept an input pair$(P_j, W_j) \in P, W$, given P is the web page and W is the set of web page categories which is assigned to a Boolean pair. The value of the Boolean pair True (T) is assigned to(P_j, W_j), if the web page P_j belongs to the web page category W_j, otherwise it is False (F) if it is not associated with it. Therefore, the goal of any web page classification system is to construct a model $\emptyset: D * C \rightarrow (T, F)$ which associates one or more categories with a web page P_j such that the result given by the model corresponds to a large extent with the $\emptyset: D * C \rightarrow (T, F)$ [4], [5]

The general problem of web classification can be divided in to three areas: classifier construction, document representation (DR) and classifier evaluation [6]. Machine Learning (ML) algorithms such as Naïve bayes, K-nearest neighbor, Decision tree, Artificial Neural Network (ANN), Support Vector Machine (SVM) and so on have been used previously by many researchers to achieve this task [7].

To achieve high classification result of the Web Page Classification (WPC) system, an excellent representation of textual data (Preprocessing/DR) should contain as much information as possible from the original document [8]. Also, the accuracy of most classification algorithms depends on the quality and size of training data which is inherently dependent on the document representation technique [9]. Several researchers have contributed to the document representation stage of the web page classification system because irrelevant and redundant features often degrade the performance of the classification algorithms both in speed and classification accuracy and also its tendency to reduce overfitting [10]. Drawing from literature, state-of-the-art DR technique's used in WPC systems are: bag of words model, Term-Frequency Inverse Document Frequency (TF-IDF), Latent Semantic Indexing (LSI), Probabilistic Latent Semantic Indexing (PLSI), Latent Dirichlet Allocation (LDA), LSI and TF-IDF, N-Gram and TF-IDF, Word2Vec and TF-IDF, TF-IDF and firefly Algorithm, Word2Vec and LDA [8], [11], [12], [13], [14], [15]. Each of these technique are fraught with one challenge or the other such as semantic mismatch and multiple meanings of word and so on.

Recently, Doc2Vec which learns document embedding's are gaining popularity because of their unique characteristics that learns continuous distributed vector representations for pieces of texts from documents. Doc2vec was created by Quoc Le and Tomas Mikolov, which was motivated by the earlier and successful release of Word2Vec by Tomas Mikolov at Google. Word2Vec learn the semantic similarity between word in vector space e.g., vec(Berlin) - vec(Germany) + vec(France) is close to vec(Paris). This suggests that distances between embedded word vectors are to some degree semantically meaningful. However, in the work of [16], they found out that word embedding's created by word2vec might perform badly on web documents with varying length. Also, they are learnt by a probabilistic language model which is not optimal for text classification problems.

In this paper, we thus propose to apply neural network language model (Doc2Vec) to solve the document representation technique of web page classification system that will capture the real semantics of web documents. The rest of this paper is structured as follows: Section 2 presents

the review of related works. In section 3, the methodology of the proposed framework is presented. Section 4, discusses the expected results and Section 5 concludes the paper.

2. RELATED WORKS

2.1. Web Page Classification

[5] proposed a hybrid document representation technique using visual content-based web page categorization with deep transfer learning and metric learning. Their work proposed a novel framework for the categorization of web pages on the basis of their visual content. They posit that previous works concentrated on the use of mining textual information and meta-data such as plain text, hyperlinks, HTML structures and so many others. These works neglect the rich information present in multi- media content on websites, which could affect the overall accuracy of the classification system. They created a pipeline which accepts a given a URL, then the system access that URL, extract all the images available on the web page, and filter those that do not contain any discriminative information. It then extracts a feature descriptor from each image such that the classification problem becomes easier over that feature space and finally analyses each feature descriptor and combine the results to produce a prediction concerning the category of the entire web page.

[17] developed a system for the categorization of Bangla web text documents. Several researchers have implemented systems for the automatic categorization of English text, however, limited studies have been carried out on the categorization of Indian language texts including Bangla. Also, their paper argues that a hybrid document representation techniques created by the addition of Inverse Class Frequency (ICF) measure to the Term Frequency (TF) and Inverse Document Frequency (IDF) methods can yield better responses in the act of feature extraction from a language like Bangla. After the features are extracted using ICF, TF-IDF, they are consequently fed in to a MultiLayer Perceptron (MLP) classifier that produces a model which can then be used to classify other Bangla text. Results obtained when compared with other classifier confirms their method produced higher accuracy in the categorization of Bangla text documents

[18] proposed a web-based classification tool for top level domain (TLD). The methodology is preceded by creating a custom made crawler that is able to crawl up to a certain number of pages starting from the main page, discarding non HTML pages, automatically discard non relevant pages such as Contacts and so on. Then using python NLTK processor to perform traditional text processing such as lemmatization, stop words removal and so on. They used a bag of words model (tf-idf) to construct the term document matrix. Then the terms are then trained using Naive Bayes and SVM classification algorithm. Experimental results shows that Naive Bayes classifier performs better than the SVM algorithm using Precision, recall and F1 score. A major gap identified in their work is that the document representation technique used was TF-IDF which is a bag of words model which does not capture semantic similarity and word order of the document being transformed [19]

[14] applied neural network model (Word2Vec) with bags of words model (tf-idf) to solve the document representation problem of web classification. Accurate Representation of documents affect the correct classification or categorization of new documents. To solve the document representation problem, they created a hybrid of Word2Vec weighted by tf-idf with stop words and tf-idf without stop words to correctly represent the feature vectors of a document. The proposed method was applied to 20 newsgroup text dataset. Experimental results show that the method performs better than tf-idf with/without stops words and word2vec with/without stop words. A major drawback in their work is that, stops words increase the dimensionality of the feature vectors which impacts badly on the classification accuracy and computational burden [8].Also the classification algorithm used was a linear SVM, other kernels such as string and RBF kernels could produce better results [20].

In the works of [8], they proposed the use of a hybrid strategy that consist of Latent Dirichlet Allocation (LDA) and Word2Vec for document representation. Word2Vec create a vector representation of the document which shows the semantic relationship between the words of the document. Euclidean distance was used to measure and interpret similarity between document and topic in sparse space. Their methods was applied to 20 News group data using SVM classifier. Results obtained shows that their proposed methods outperforms earlier methods such as TF-IDF+ SVM, Word2Vec + SVM, LDA + SVM. One of the major drawback of their method is that improper calibration of the LDA parameters (e.g. number of topics, hyper-parameters), could potentially lead to sub-optimal results as most of the parameters for the LDA are imported from natural language community [21].

2.2. Doc2Vec

An extension of word2vec that encodes entire documents as opposed to individual words. In this case, a document can be a sentence, a paragraph, an article, an essay, and so on. Like word2vec, doc2vec (sometimes referred to as paragraph vectors) relies on a supervised learning task to learn distributed representations of documents based on contextual words. Doc2vec is also a family of algorithms, whereby the architecture will look extremely similar to the CBOW and skip-gram models of word2vec. The two models of Doc2Vec are Distributed-Memory Model (DMM) and Distributed Continuous Bag of Word Model (DCBOW). The distributed-memory model algorithm tries to predict a focus word given its surrounding context words but with the addition of a paragraph ID. The paragraph ID can be thought of as another individual contextual word vector that helps with the prediction task but is constant throughout what we consider to be a document. The 2nd doc2vec algorithm is modeled after the word2vec skip-gram model, with one exception--instead of using the focus word as the input, we will now take the document ID as the input and try to predict randomly sampled words from the document. That is, we will completely ignore the context words in our output altogether. Figure 1 below shows the architecture of Doc2Vec framework

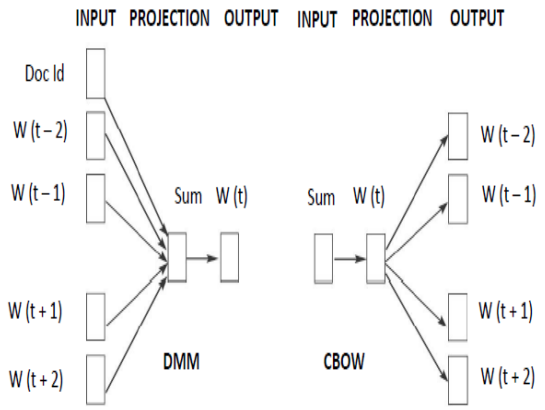

Figure 1: Architecture of a Doc2Vec Framework model [22]

3. METHODOLOGY

This section describes the proposed neural network language model for document representation using Doc2Vec algorithm. The following will show the different phases in our proposed methodology in capturing the semantic similarity of web documents.

3.1 Data Collection

To collect dataset for our proposed work, we design a custom made web crawler using scrappy tool (A python-based library for web crawling). The web crawler will be used to crawl on various subject from a popular blog site in Nigeria such as Pulse.ng or Nairaland etc. The subject for the web pages to be crawled will be in the following categories: entertainment, fashion, politics, religion and sports. The following categories were carefully chosen after surfing other blogs in Nigeria such as Lindaikeji.com, naij.com, vanguard.com, punchng.com. Python based Scrappy toolkit was used for web scrapping and extracting data from the blog site. It also allows the content extracted to be saved to a csv file. Also Beautiful soup python framework was used to pull out the real data from the HTML and XML files. Four annotators were used to label the blog posts.

Figure 2 below shows the distribution of corpus for each category

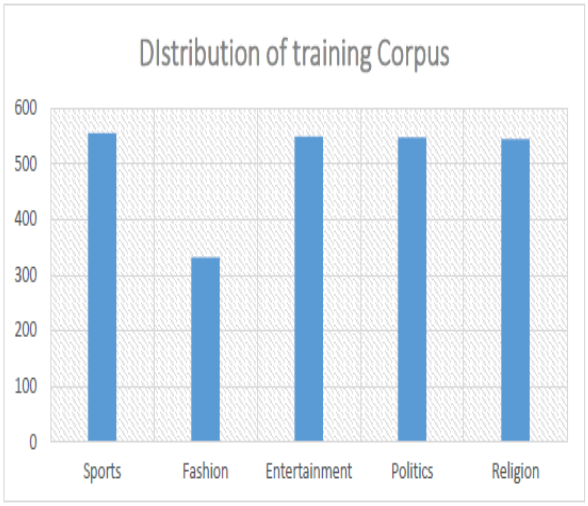

Figure 2: Distribution of Corpus

3.2 Pre-Processing

After crawling and annotating the blog post. They are used to create a corpus which is saved in a .csv file. Then standard text preprocessing are applied to the corpus of training dataset such html segmentation, tokenization, Lowercase conversion, Stop-words removal, stemming, lemmatization, and POS removal. Python NLTK module for tokenization, lowercase conversion, stop words removal, POS removal. Python module snowball is used for further stemming. Also, python module TreebankWordTokenizer was used for enhanced tokenization to break all words in to individual text.

3.3 Overview of the Methodology

The methodology proceeds by taking the already pre-processed data set and adding a label at the end of each blog post of each category to serve as the paragraph ID. This is then passed in to the doc2Vec algorithm DCBOW model. The doc2vec model is then created for that category. The default parameters of the model was used with alpha=0.025, vector_size=5, window=2, min_count=1, workers=4. 66% of the corpus dataset was used for training and 34% for testing. After creating the word vectors for each category, a matrix containing the word_vectors and a label for all the categories was created and saved in a csv file. Then matrix containing all categories is then passed in to a machine learning algorithm based classifier. The python-based framework genism module was used to implement the doc2vec model. The classifier for the Web Page Classification (WPC) system will be developed using Support Vector Machine (SVM) with Radial Basis Function Kernel (RBF). One versus All (1VA) version of SVM will be used for web pages that belong to more than one category. Use Scikit-learn to implement SVM 1VA classifier.

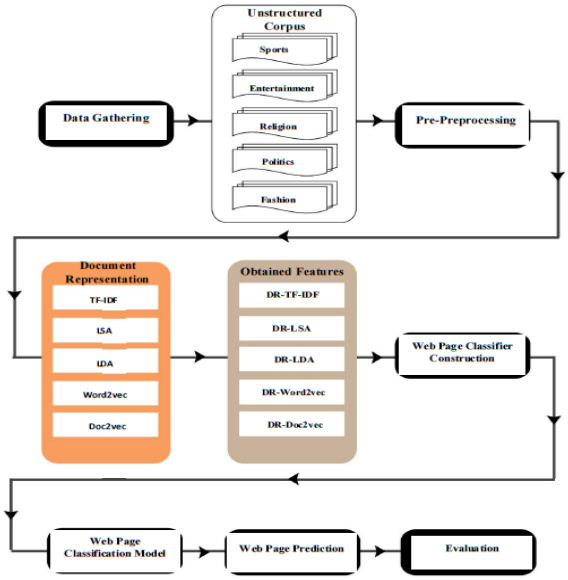

Figure 3: Overview of the Methodology

3.4 Evaluation of the Proposed System

The proposed system was validated by diving the corpus in to training and test samples. The system will be evaluated in terms of accuracy, precision, recall and F1 ratio. To investigate the performance of the proposed Doc2Vec model for document representation, we compared Doc2Vec with other TF-IDF, LSA, LDA and Word2Vec.

Table1: Classification report of the TF-IDF model

.Category	Precision	Recall	F1 score	Support
Entertainment	0.84	1	0.94	22
Fashion	1	0.96	0.98	25
Politics	0.96	0.96	0.96	26
Religion	1	0.93	0.96	28
Sports	1	1	1	24
Avg / total	0.89	0.88	0.91	125

Table 2: Confusion Matrix of the TF-IDF model

Category	Entertainment	Fashion	Politics	Religion	Sports
Entertainment	16	0	0	1	5
Fashion	2	17	1	1	4
Politics	1	0	22	3	0
Religion	5	0	5	18	0
Sports	6	0	0	1	17

Table 3: Classification report of the LDA model

Category	Precision	Recall	F1 score	Support
Entertainment	0.89	0.77	0.83	22
Fashion	1	0.84	0.91	25
Politics	0.8	0.92	0.86	26
Religion	0.82	1	0.9	28
Sports	0.9	0.79	0.84	24
Avg / total	0.88	0.87	0.87	125

Table 4: Confusion Matrix of the LDA model

Category	Entertainment	Fashion	Politics	Religion	Sports
Entertainment	18	0	0	1	3
Fashion	1	18	1	1	4
Politics	2	0	22	2	0
Religion	5	0	5	18	0
Sports	6	0	0	1	17

Table 5: Classification report of the Word2Vec model

Category	Precision	Recall	F1-score	Support
Entertainment	0.76	0.82	0.67	22
Fashion	1	0.68	0.81	25
Politics	0.78	0.81	0.79	26
Religion	0.75	0.64	0.69	28
Sports	0.72	0.75	0.73	24
Avg / Total	0.77	0.74	0.74	125

Table 6: Confusion Matrix of the Word2Vec model

Category	Entertainment	Fashion	Politics	Religion	Sports
Entertainment	17	0	0	1	4
Fashion	1	19	1	1	3
Politics	2	0	21	3	0
Religion	5	0	5	18	0
Sports	6	0	0	1	17

Table 7: Classification report of the Doc2Vec model

Category	Precision	Recall	F1 score	Support
Entertainment	**0.90**	**0.94**	**0.92**	**79**
Fashion	**0.88**	**0.98**	**0.93**	**53**
Politics	**0.92**	**0.92**	**0.92**	**93**
Religion	**0.93**	**0.88**	**0.90**	**88**
Sports	**0.96**	**0.92**	**0.94**	**105**
Avg / total	**0.92**	**0.92**	**0.92**	**418**

Table 8: Confusion Matrix of the Doc2Vec model

Category	Entertainment	Fashion	Politics	Religion	Sports
Entertainment	**19**	**0**	**0**	**1**	**2**
Fashion	**1**	**20**	**1**	**1**	**2**
Politics	**1**	**0**	**24**	**1**	**0**
Religion	**5**	**0**	**3**	**20**	**0**
Sports	**3**	**0**	**0**	**1**	**20**

Table 9: Performance Measure Varying the Number of features

Model	Accuracy
TF-IDF	**77.4**
LDA	**87.2**
Word2Vec	**92.2**
Doc2Vec	**98.7**

4. FINDINGS AND DISCUSSION

Findings 1: It is obvious that the model that produced the best accuracy is Doc2Vec with a score of 98.7 followed by word2Vec with a score of 92.2 and LDA at 87.2 and TF-IDF at 77.4.This shows that Doc2Vec produced better semantics about the category of the web pages than the other models.

Findings 2: On computing the confusion matrix of each model, it is apparent that the Doc2Vec model predicted the correct category for most of the web documents followed by

Word2Vec.This shows that doc2vec is able to represent the document in a way that the classifier can correctly classifier the right category the web document belongs to

Findings 3: On plotting the accuracy of the models with a varying number of features, TF-IDF perform badly as the features increases, LDA performs moderately as the features scales up. Word2Vec scales well with increase in the number of features. Doc2Vec increases performance as the feature set increases, scaling well with the high dimensional nature of web document

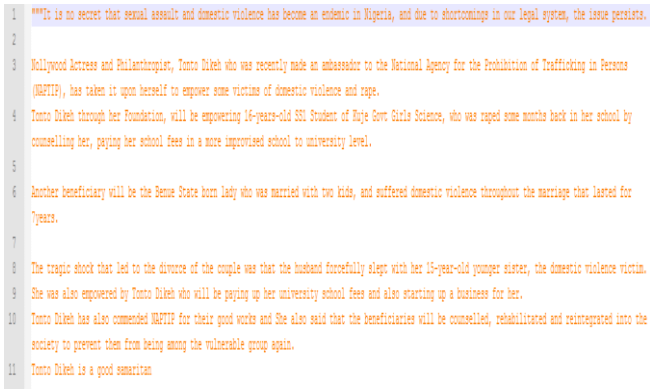

Figure 4: Sample Blog Post

Figure 4 above, shows an original blog post from the entertainment corpus. To further investigate the semantic relatedness amongst the document vectors created by doc2vec, the value of the cosine similarity was compared with other models (TF-IDF, LSI, PLSI, LDA and Word2Vec). Doc2vec performs optimally.

Table 10: Cosine Similarity of Various Models

Model	Cosine Similarity
Doc2Vec	0.9876
TF-IDF	0.8862
LSI	0.7862
PLSI	0.8526
LDA	0.8867
Word2Vec	0.9687

5. CONCLUSION

The proposed Doc2Vec approach obviously provides a better document representation technique for web page classification i.e. the document embedding's used by Doc2Vec model can be used to obtain better semantics of large block of text such as that of blogs. The proposed system will help search engines to improve the quality of web directories in organizing the vast amount of blogs and websites into hierarchical collections. Also, the proposed system will also help users to retrieve the accurate information needed by end users to avoid information over load.

6. REFERENCES

[1] Azam, N., & Yao, J. (2012). Comparison of term frequency and document frequency based feature selection metrics in text categorization. Expert Systems with Applications, 39(5), 4760-4768.

[2] Tragha, A. (2019). Machine Learning for Web Page Classification: A Survey. International Journal of Information Science and Technology, 3(5), 38-50.

[3] Virik, M., Simko, M., & Bielikova, M. (2017). Blog style classification: refining affective blogs. Computing and Informatics, 35(5), 1027-1049.

[4] Karima, A., Zakaria, E., Yamina, T. G., Mohammed, A. A. S., Selvam, R. P., & Venkatakrishnan, V. (2012). Arabic text categorization: a comparative study of different representation modes. Journal of Theoretical and Applied Information Technology, 38(1), 1-5.

[5] Lopez-Sanchez, D., Arrieta, A. G., & Corchado, J. M. (2019). Visual content-based web page categorization with deep transfer learning and metric learning. Neurocomputing, 338, 418-431.

[6] Khan, A., Baharudin, B., Lee, L. H., & Khan, K. (2010). A review of machine learning algorithms for text-documents classification. Journal of advances in information technology, 1(1), 4-20.

[7] Fatima, S., & Srinivasu, B. (2017). Text Document categorization using support vector machine.

[8] Ma, S., Zhang, C., & He, D. (2016). Document representation methods for clustering bilingual documents. Proceedings of the Association for Information Science and Technology, 53(1), 1-10.

[9] Dey Sarkar, S., Goswami, S., Agarwal, A., & Aktar, J. (2014). A Novel Feature Selection Technique for Text Classification Using Naïve Bayes. International Scholarly Research Notices, 2014.

[10] Alamelu Mangai, J., Santhosh Kumar, V., & Sugumaran, V. (2010). Recent Research in Web Page Classification–A Review. International Journal of Computer Engineering & Technology (IJCET), 1(1), 112-122.

[11] Chen, R. C., & Hsieh, C. H. (2006). Web page classification based on a support vector machine using a weighted vote schema. Expert Systems with Applications, 31(2), 427-435

[12] Zhang, W., Yoshida, T., & Tang, X. (2011). A comparative study of TF* IDF, LSI and multi-words for text classification. Expert Systems with Applications, 38(3), 2758-2765.

[13] Karima, A., Zakaria, E., Yamina, T. G., Mohammed, A. A. S., Selvam, R. P., & VENKATAKRISHNAN, V. (2012). Arabic text categorization: a comparative study of different representation modes. Journal of Theoretical and Applied Information Technology, 38(1), 1-5.

[14] Lilleberg, J., Zhu, Y., & Zhang, Y. (2015, July). Support vector machines and word2vec for text classification with semantic features. In Cognitive Informatics & Cognitive Computing (ICCI* CC), 2015 IEEE 14th International Conference on (pp. 136-140). IEEE.

[15] Raj, A. J., Francis, F. S., & Benadit, P. J. (2016). Optimal Web Page Classification Technique Based on Informative Content Extraction and FA-NBC. Computer Science and

[16] Huang, C., Qiu, X., & Huang, X. (2014). Text classification with document embeddings. In Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data (pp. 131-140). Springer, Cham.

[17] Dhar, A., Dash, N. S., & Roy, K. (2018, January). Categorization of bangla web text documents based on TF-IDF-ICF text analysis scheme. In Annual Convention of the Computer Society of India (pp. 477-484). Springer, Singapore.

[18] Deri, L., Martinelli, M., Sartiano, D., & Sideri, L. (2015, November). Large scale web-content classification. In Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K), 2015 7th International Joint Conference on (Vol. 1, pp. 545-554).

[19] Singh, K. N., Devi, H. M., & Mahanta, A. K. (2017). Document representation techniques and their effect on the document Clustering and Classification: A Review. International Journal of Advanced Research in Computer Science, 8(5).

[20] Nayak, J., Naik, B., & Behera, H. (2015). A comprehensive survey on support vector machine in data mining tasks: applications & challenges. International Journal of Database Theory and Application, 8(1), 169-186.

[21] Dit, B., Panichella, A., Moritz, E., Oliveto, R., Di Penta, M., Poshyvanyk, D., & De Lucia, A. (2013, May). Configuring topic models for software engineering tasks in tracelab. In Traceability in Emerging Forms of Software Engineering (TEFSE), 2013 International Workshop on (pp. 105-109). IEEE.

[22] Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In International conference on machine learning (pp. 1188-1196).