EMPLOYING THE CATEGORIES OF WIKIPEDIA IN THE TASK OF AUTOMATIC DOCUMENTS CLUSTERING

Abdullah Bawakid

Faculty of Computing and Information Technology, University of Jeddah, Jeddah, Saudi Arabia

ABSTRACT

In this paper we describe a new unsupervised algorithm for automatic documents clustering with the aid of Wikipedia. Contrary to other related algorithms in the field, our algorithm utilizes only two aspects of Wikipedia, namely its categories network and articles titles. We do not utilize the inner content of the articles in Wikipedia or their inner or inter links. The implemented algorithm was evaluated in an experiment for documents clustering. The findings we obtained indicate that the utilized features from Wikipedia in our framework can give competing results especially when compared against other models in the literature which employ the inner content of Wikipedia articles.

KEYWORDS

Wikipedia, Documents Clustering,Wikipedia Categories

1. INTRODUCTION

The task of automatic documents clustering is an important one in the data mining domain, especially with the abundance of data and its continuous expansion on the web.An example application for this is the updates for news services provided by different operators on the internet. Users usually subscribe first to the services and topics they are interested in while the service provider feeds the users with live news according to their subscriptions. A user can be overloaded with the news items that are received which can be redundant, related or about the same topic. One way to help the user in organizing the received items is to have this clustered. In other words, it would be useful to device a way to automatically cluster the received news items based on the themes or topics in each item, hence the need for automatic documents clustering.

In this paper we describe a framework we designed and adapted its algorithm for the task of automatic text documents clustering. The main algorithm in the framework utilizes an external knowledge repository for enriching the representation of text documents. The external repository we employ is the largest known encyclopaedia to date, namely Wikipedia. In contrast to the work performed previously in the literature, we utilize only two aspects from Wikipedia in our algorithm: the articles titles and the categories network. We do not employ the internal content of each article in Wikipedia or its internal or external links.To evaluate our system, we utilized an external dataset to run an experiment for automatic documents clustering. We provide details in this paper about the experiment and our findings.

DOI: 10.5121/ijci.2015.4304

The remaining of this paper is organized as follows: In the next section we give an overview about the related work in the field. In section 3, we give more details about the framework we implemented and the different stages involved when extracting the features from Wikipedia. In section 4 we explain the different variations of our system. We also discuss the experiment we ran and the results of our evaluation. In Section 5 we summarize our findings and potential future work.

2. BACKGROUND AND RELATED WORK

The documents clustering task has had considerable studies in the literature especially in the past two decades. Many of the techniques covered in the literature rely on the explicitly found terms in the test documents that are to be clustered. For example, the work in [1]utilizes the keywords and phrases that exist in the test document collection. They apply statistical techniques on the discovered key phrases to establish clusters of similar documents. Similarly, the work in [2] and [3] have shown promising results with the usage of a bag-of-words model for documents clustering. One drawback when using a bag-of-words model is that it merely captures explicitly mentioned words in the test documents. It does not necessarily take into account the different forms or shapes for the document terms. For instance, the words 'run' and 'ran' would be treated as two separate and unrelated terms.

Other methodologies in the literature looked at the text representation problem for clustering from another perspective. They devised a technique for assigning weights to the different terms and phrases within text documents with a feature weighing model such as the TF-IDF model[4] referring to term frequency-inverse document frequency. With this model, the frequencies of a term within each document in addition to its commonness among the documents are both taken into account when giving a weight to each term. After representing the documents terms with weighted vectors, a supervised clustering algorithm is applied such as Support Vector Machines (SVM) in [5], Naïve Bayes in[6], or Decision Trees in [7] and [8].

One of the shortcomings for feature weighing based approaches covered above is that they are all supervised and require training data. Hence, the performance of their algorithms is dependent on the quality of the training data at hand. The amount of noise that may exist in either the training data or actual test documents to be clustered is also a major factor on the overall performance of system.

Other systems in the literature took a different approach by enriching the representation of text documents through inflating them with additional text. For example, the work in [9] and [10] attempted to inflate the test documents text with the results obtained from Google search. They used the statistics obtained from Google search result to determine the similarity of any two text fragments. The main advantage of this method is that it does not require pre-processed ontologies or data repositories. However, while this method may be useful for short text, it may not be practical for documents with long text. Another drawback for this is the ambiguity of some text fragments that are fed to the search engine. For ambiguous phrases, the search results may produce even more ambiguous results. For these cases, human intervention may be required to remove the ambiguity before feeding search engines with the search queries.

Another approach that was studied in the literature is the usage of external ontologies for enriching text representation. For example, the systems described in [11] and [12]have explored the usage of WordNet for enriching text representation before clustering the documents. The systems attempted to compute the semantic similarity between different text fragments in the documents to determine the overall similarity between every two test documents in the collection

to be clustered. The conceptual and lexical relations within WordNet were employed for this purpose. Other ontologies such as OpenCyc and SUMO have been employed in the literature, too[13]. Among the major drawbacks for the mentioned ontologies is their limitation in content and expandability. These ontologies were constructed manually by experts and their content is limited and does not span every domain. New and emerging concepts are not necessarily covered in these ontologies, too.

Several subsequent studies in the literature focused on using open world data repositories instead. An example for such a repository is Wikipedia which is known to be rich in content and is also expandable. In [14] and [15] , Wikipedia was used as a mean to enrich the representation of text documents. A connection between any two documents is established by merely examining the overlap of the Wikipedia concepts this exist between the two documents. The semantic similarity or relatedness between concepts is not taken into account in their systems.In an attempt to bypass this paper, other methodologies as in [16] and [17] devised means to compute the semantic relatedness between concepts with the aid of Wikipedia. These methodologies utilized many aspects from Wikipedia including its anchor links, inner text content for each article, titles redirect links and the categories network. The performance for systems employing these methodologies was found to be encouraging.

In this paper, we describe an algorithm that utilizes Wikipedia for enriching a document representation and also clustering text documents. In contrast to others mentioned in the literature, our method differs in that it does not merely look at the overlap of concepts explicitly mentioned in text documents, but also takes into account the relatedness between these concepts. This semantic relatedness between inter-document concepts is indirectly taken into account through the usage of the term-categories vector we prepared from Wikipedia. Furthermore, our method differs from those in the literature from another perspective: it is unsupervised and only uses two aspects from Wikipedia, namely the articles titles and the categories network. This makes it faster to run, implement, and does not require too much processing or memory resources to run.

3. FRAMEWORK

The main algorithm that is utilized by our framework requires the titles of the articles in Wikipedia in addition to its categories network. Before we are able to use the Wikipedia dump we downloaded from Wikipedia's website, we have to apply to it several steps to extract the required features. The result of these steps is the features that will be employed in our framework for the task of documents clustering.We provide a description for the main stages involved to generate the required features in the following subsections:

3.1. Preprocessing Wikipedia

As a first stage, we remove the undesired data from Wikipedia articles which are not employed by the main algorithm. This was applied in the implemented framework by retaining only the title of each article along with the its attached categories. We discard the inner content of each article including its text, images, tables and any other Meta tags the article may have. As for the categories network, we remove the too broad categories which contain more than 2000 articles. We also remove the too narrow categories containing less than 10 articles. Additionally, we removed non-useful categories that are mainly used for maintenance or administrative purpose within Wikipedia such as those containing only numbers as in "1940s" (referring to years) and "protected images".

3.2. Constructing Term-Categories Vector

The goal of the term-categories vector is to establish a relationship between each term and the different categories that belong to Wikipedia. This relationship is presented in the form of a weight resembling the relatedness strength of a term to a Wikipedia category. The term-categories vector is constructed by applying the following formula:

$$w_t = \frac{1}{|a_t|} \times log \frac{|C|}{|C_t|} \quad (1)$$

In the above formula, we have $w_t$as the term weight for the term t,$|a_t|$as the total articles titles number which contain the term$t$, $|C|$as the total Wikipedia categories number, and $|C_t|$is the number of Wikipedia categories which contain the term$t$. In the next section we describe how to employ the obtained term weight when processing the test documents.

3.3. Processing Test Text Documents

We begin dealing with every test document $d$that needs to be clustered by first computing the frequency for every term t that belongs to $d$and store the result in wf_t. Afterwards, we attempt to obtain the Wikipedia categories which represent $d$best by the inferred relationship among the titles of Wikipedia articles and the categories linked to every article. We cover this step byimplementing the equation that follows:

$$w_a = \text{sum}_{t \to d}(wf_t \times w_t) \times \frac{|a_d|}{|a|} \quad (2)$$

In equation 2, we have $w_a$referring to the article title weight, wf_treferring to the t word frequency, $|a|$referring to the total words numberwhich are present in the article title a, $|a_d|$referring to the total terms number that exist in both the document $d$and the title of the article a.
It should be noted that we also considered the alternative titles for each article in Wikipedia which are also called “alternative links”. In Wikipedia, each article may have one title or more alternative titles. When applying our algorithm, we assign a weight to the main title in addition to the alternative ones. However, for each article we always consider the title with the maximum weight in the implementation of our algorithm.

3.4. Documents Clustering

At this stage, a weight $w_a$should be attached to each Wikipedia article title a. This weight should reflectthe relevance of a to the text document d. Afterwards, we compute a score for every Wikipedia category which reflects how representative the category is with respect to document d. This is applied through the aggregation of the weights of the articles titles $w_a$by taking their sum intow_c. The higher the computed score for a category, the more representative it is to d. We form a ranked list of the obtained scores in a descending order and choose the top P categories to be the best representative for d.

After generating the top P categories for each document, we form the clusters of text documents by utilizing their most representative categories. At the beginning, we have a number of clusters that is equal to the total number of test documents. We compute the cosine distance between the

most representative categories vectors for each document against the rest of the documents in the data set. When the distance is found to be less than a previously defined threshold *K* we consider the document to be part of the cluster which shares the least distance with.

4. EXPERIMENTS AND EVALUATIONS

We created different variations of the framework for testing its performance. In addition to the main version described above (we call hereafter *V1*), we created a version in which stemming was applied in the pre-processing stage as well as when processing text documents. The titles of Wikipedia articles were stemmed in addition to the alternative articles titles too. We refer to this variation as *V2*.

We also created another variation in which we modified the formula used for computing the weight of each article titleto become as follow:

$$w_a = \text{sum}_{t \to d}(wf_t \times w_t) \times \frac{1}{|L_a|} \times \frac{|a_d|}{|a|} \quad (3)$$

In the above equation, we have $|L_a|$which refers to the number of articles sharing the same title as that of article *a*. This addition is important especially with the variant in which stemming was applied. It is therefore useful to give more weight to articles titles which link to the least amount of articles. In the same time, the weight of articles titles which point to many articles will be less with this new feature, effectively giving the rest of the features in equation (3) more weight than this particular feature. We refer to this variation of our framework as *V3*.

In order to evaluate the performance of the framework in a clustering task, we utilized the 20-newsgroups (20NG) dataset for this purpose. We employed all of the 19,997 documents and the 20 classes present in 20NG in our testing. The evaluation was performed by creating four different groups for each class in the dataset by randomly selecting 100 documents for each group from each class. The evaluation was completed separately for each group and the average score for all groups are collectively taken as a representative score for the whole dataset.

We used two evaluation metrics in the experiment we ran, namely F-Score[18], Inverse Purityand Purity[19]. The former metric is commonly used as a combination for precision and recall. The later metric considers all sample documents in a cluster to belong to the dominant class in that cluster. Inverse Purity focuses on maximum recall for each cluster in each class.

As a baseline, we also included a run in our experiment in which the bag of words (BOW) modelwas implemented. The explicitly mentioned terms in each document are used to form a representative words vector for the document. Clustering takes place by comparing the explicitly mentioned terms within each document (in the words vector) against the rest in the dataset and finding the documents with maximum exact matches.

Table 1. Evaluation Results

	Purity	**Inverse Purity**	**F-Score**
BOW (Baseline)	0.130	0.101	0.141
V1	0.113	**0.107**	0.121
V2 (Stem)	**0.131**	0.100	0.139
V3	**0.132**	**0.119**	**0.146**

The obtained results from our evaluation are reported in Table 1. In that table, *BOW* refers to our implementation of the bag-of-words model, *V1* refers to our original framework and its implementation as described above, V2 refers to the version of our framework where stemming was applied while *V3* is the version in which formula 3 was adapted. It can be noted from the obtained results that *V3* obtained the best results for all 3 metrics. *V1* obtained a better Inverse Purity than the baseline but failed to provide improvements with the other two metrics. In the second variation of our framework in which stemming was applied, namely *V2*, the results we obtained were relatively close to the baseline. The improvement made to *V2* by introducing a feature that gives more emphasis on titles pointing to the least amount of articles as reflected in formula 4 resulted in *V3*. This added feature proved to be effective in improving the obtained scores for our framework in all three metrics.

5. CONCLUSION

In this paper, we described a framework we developed and adapted to be used in the task of automatic documents clustering. The framework leverages specific aspects of Wikipedia, namely its categories network and articles titles, for implementing an unsupervised automatic documents classifier. The algorithm developed for this task involves the creation of a term-categories vector. This vector defines how strongly a term is related to all the categories in Wikipedia. We explained how this vector was generated and utilized in our algorithm.

The developed framework was tested in the task of documents clustering by utilizing the 20NG dataset. Our findings from the evaluation results indicate that the features generated mainly from the categories network of Wikipedia in addition to its articles titles can be sufficient for developing systems with competing evaluation results in the task of automatic documents clustering. It is not always necessary to employ the inner content of Wikipedia articles for developing competitive frameworks as illustrated in the results obtained with the baseline we utilized. We find the reported findings in this paper encouraging for further enhancing the implemented algorithm and also adapt it to be applied in several related applications including documents labelling and information retrieval. We intend to explore these options next.

REFERENCES

[1] J. Sivic and A. Zisserman, “Video Google: a text retrieval approach to object matching in videos,” in Ninth IEEE International Conference on Computer Vision, 2003. Proceedings, 2003, pp. 1470–1477 vol.2.

[2] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, “Visual categorization with bags of keypoints,” Workshop Stat. Learn. Comput. Vis. ECCV, vol. 1, no. 1–22, pp. 1–2, 2004.

[3] Y. Zhang, R. Jin, and Z.-H. Zhou, “Understanding bag-of-words model: a statistical framework,” Int. J. Mach. Learn. Cybern., vol. 1, no. 1–4, pp. 43–52, Aug. 2010.

[4] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. New York, NY, USA: Cambridge University Press, 2008.

[5] L. Ping, Z. Chun-Guang, and Z. Xu, “Improved support vector clustering,” Eng. Appl. Artif. Intell., vol. 23, no. 4, pp. 552–559, Jun. 2010.

[6] K. Sarkar, M. Nasipuri, and S. Ghose, “Machine Learning Based Keyphrase Extraction: Comparing Decision Trees, Naïve Bayes, and Artificial Neural Networks,” J. Inf. Process. Syst., vol. 8, no. 4, pp. 693–712, Dec. 2012.

[7] M. K. Saad and W. Ashour, “Arabic text classification using decision trees,” in Proceedings of the 12th international workshop on computer science and information technologies CSIT ‘2010, Moscow–Saint-Petersburg, Russia, 2010.

[8] M. Alruily and M. Alghamdi, "Extracting information of future events from Arabic newspapers: an overview," in Semantic Computing (ICSC), 2015 IEEE International Conference on, 2015, pp. 444–447.
[9] "Measuring Semantic Similarity Between Words Using Web Search Engines," in Proceedings of the 16th International Conference on World Wide Web, New York, NY, USA, 2007, pp. 757–766.
[10] W.-T. Yih and C. Meek, "Improving Similarity Measures for Short Segments of Text," in Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2, Vancouver, British Columbia, Canada, 2007, pp. 1489–1494.
[11] J. Sedding and D. Kazakov, "WordNet-based Text Document Clustering," in Proceedings of the 3rd Workshop on RObust Methods in Analysis of Natural Language Data, Stroudsburg, PA, USA, 2004, pp. 104–113.
[12] D. R. Recupero, "A new unsupervised method for document clustering by using WordNet lexical and conceptual relations," Inf. Retr., vol. 10, no. 6, pp. 563–579, Oct. 2007.
[13] S. Bloehdorn and A. Hotho, "Boosting for Text Classification with Semantic Features," in Advances in Web Mining and Web Usage Analysis, 2006, pp. 149–166.
[14] A. Huang, D. Milne, E. Frank, and I. H. Witten, "Clustering Documents with Active Learning Using Wikipedia," in Eighth IEEE International Conference on Data Mining, 2008. ICDM '08, 2008, pp. 839–844.
[15] X. Hu, N. Sun, C. Zhang, and T.-S. Chua, "Exploiting Internal and External Semantics for the Clustering of Short Texts Using World Knowledge," in Proceedings of the 18th ACM Conference on Information and Knowledge Management, New York, NY, USA, 2009, pp. 919–928.
[16] P. Sorg and P. Cimiano, "Exploiting Wikipedia for cross-lingual and multilingual information retrieval," Data Knowl. Eng., vol. 74, pp. 26–45, Apr. 2012.
[17] S. Banerjee, K. Ramanathan, and A. Gupta, "Clustering Short Texts Using Wikipedia," in Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 2007, pp. 787–788.
[18] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in In KDD Workshop on Text Mining, 2000.
[19] Y. Zhao and G. Karypis, "Criterion Functions for Document Clustering: Experiments and Analysis," 2002.