Clustering Documents Using Wikipedia

Abstract:

clustering of documents using bag of words model involves measuring similarity between them by co-occurrence statistics. Clustering algorithms which use BOW model can relate documents that use identical terminology. While the semantic relations are ignored. Recent studies had revealed that incorporating supervision improves clustering performance. Wikipedia has been utilized as a resource for many text mining problems. I had used wikipedia for getting the semantic relatedness and used it for clustering documents. Semantic relatedness is a measure of how related two or more concepts are. At first, I had taken a set of files from different group of data and created a master document. I had preprocessed this document by using parts of speech tagger and had took only noun and adjectives as input. After getting the refined data I had applied it to dbscan algorithm to get the clustered concepts. Using these clustered concepts generate a mxn matrix where n is the number of clustered concepts and m is the number of files. For each file calculate the weight of the clustered concept. Now, apply this matrix to k-means algorithm. This will give output as a clustered documents.

Introduction:

The goal is to present a clustering technique using Wikipedia. Clustering of document involves grouping the similar type of documents based on some criteria. In the methodology presented in this paper, I had used semantic relatedness . The benefit of using such concept is to group documents in the same cluster even if they do not contain common keywords, but still possess the same sense. Existing clustering techniques cannot perform this sort of discovery or do this work only to a limited degree. Developing algorithm that discover all frequently occurring subgraphs in a large graph database is particularly challenging. I had used dbscan to discover the subgraphs. After discovering the subgraphs the documents are clustered using kmeans algorithm.

Related work:

similar work has been done by Anna Huang[1] for clustering of documents using active learning. They had used cop k-means which will check for new constraints and reloop the process. Their method involved identifying the possible candidate words

by mapping each word or phrases with the anchors present in wikipedia. They had used Mihalcea and Csomai's keyphraseness feature to discard all the unwanted terms. In another research Recupero and Hotho had also clustered documents using wikipedia to get a semanticlly enriched document but they used wordnet as knowledge instead of wikipedia.

Using Wikipedia to predict semantic relatedness between concepts has recently attracted a significant amount of interest. Alternatives to the measure from Milne and Witten which uses hyperlinks to measure the similarity include WikiRelate!, which utilizes the Wikipedia category structure to compute similarity between articles; explicit semantic analysis from Gabrilovich and Markovitch, where sophisticated content analysis is used; and Wang et al.'s work, which models relatedness between two concepts as a linear combination of the similarity between multiple aspects.

Method :-

we constructed a concept graph from the complete corpus using semantic relatedness. 25 documents from 20newsgroups collection 5 of a kind. Load the wikipedia database into mysql. Spliting the text to know sentence boundary. I had used Pos tagger and chunker (Conditional Random Fields) for Extracting only noun and adjective phrases. Now, I had eliminated all the duplicates from words I got.

Mapping adjective and noun phrases to concepts in wiki using DBSCAN Build a n*n Matrix using semantic relatedness. where n is the number of concepts. The DBSCAN algorithm was first introduced by Ester, et al and relies on a density-based notion of clusters. Clusters are identified by looking at the density of points. Regions with a high density of points depict the existence of clusters whereas regions with a low density of points indicate clusters of noise or clusters of outliers. This algorithm is particularly suited to deal with large datasets, with noise, and is able to identify clusters with different sizes and shapes. In our implementation, it takes input as a semantically related works with the probability of occurance between them.

Example: lets say x,y and z are concepts in our documents then the matrix looks like:

	X	Y	Z
X	1	0.7	0.3
Y	0.7	1	0.4
Z	0.3	0.4	1

output will cluster the above concepts into set of clusters like X and Y together.

After getting the clustered concepts from the dbscan. I had considered all of these concepts removing the noise part. Now with these clustered concepts calculate the weight of them in each document. I had used term frequency to calculate the weight of each clustered concept. This will generate a m*n matrix where m is the number of files and n is the number of clustered concepts. The m*n matrix would be as follow

	{X,Y}	{A,B,C}	{E,F}
file1	6	3	2
file2	5	2	5
file3	5	4	3
file4	1	2	1

I had gave the above matrix as input to kmeans algorithm. K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters . The main idea is to define centroids, one for each cluster. Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids. When all objects have been assigned, recalculate the positions of the K centroids. Continue this process until it reaches a equilibrium point and the change in centroid values are minimum. The output would be clustered documents.

Evaluation

I had considered 20newsgroups as the corpus. From the corpus I took 25 documents from the collection 5 of a kind. From the 5 types of data I took one of the document types were about computer graphics, one of them was about alt.atheism and remaining were about comp.windows x, comp.mac hardware and sci.space. I had cross-checked these documents with the result I got.

Results

After getting the output form kmeans algorithm. I had evaluated with the corpus I took .Experiment with 25 documents had clustered 13 documents properly. Most of the output I got from dbscan included computer related terms. As I took most of the data from computer related documents. One of the cluster I got had 8 documents which were related to computers.

Conclusion and Future work:

In my current project I had used parts of speech tagger to preprocess data. I would like to add up the Mihalcea and Csomai's[3] keyphraseness feature for candidate identification. the probability of a word to become concept is measured as x(a)/x(a)+y(a). where x(a) is the number of wikipedia articles in which the word a is used as article and y(a) is the number of wikipedia articles in which it is utilized in any form. This will reduce many of ambiguous words like hot dog and south africa. I will also try to use some more preprocessing steps and also large corpus to get the exact result.

References :-

1 Clustering Documents with Active Learning using Wikipedia
Anna Huang, David Milne, Eibe Frank, Ian H. Witten

2 GDClust: A Graph-Based Document Clustering Technique
M. Shahriar Hossain, Rafal A. Angryk

3 R. Mihalcea and A. Csomai. Wikify! linking documents to encyclopedic knowledge.