# CLUSTER-BASED RETRIEVAL USING LANGUAGE MODELING APPROACH: AN EXPLANATION

Gloria Virginia[1]

## 1. Introduction

Nowadays, one of the demands for computer system is capability to process text and natural language automatically. Consequently, the development of algorithms that enable computers to do such task has been one of the great challenges. Hence, any substantial progress in this domain will have a strong impact on numerous applications ranging from information retrieval, information filtering, and intelligent agents, to speech recognition, machine translation, and human-machine interaction [10].

Information retrieval is a task to retrieve relevant documents in response to a query by measuring similarity between documents in repositories and the query. In recent years, the meaning of the term 'similar' between documents and query has been developed. At first, a document is judged similar with the query merely based on lexical matching of the word between documents and query. Now, the term 'similar' is expanded to the meaning of the query. It means that the query is not necessarily expressed in the document, to come up with the judgment that a document is similar with the query.

In this paper I'm going to explain how it can be done, that the query is not necessarily expressed in the document, in text retrieval. But before it, I'll briefly describe some methods I used here. I'll close this paper by summarizing the explanation.

## 2. Theories

### a. Clustering

Clustering is an assignment to group objects or elements by similarity. In document clustering, the objects are documents. It assigns each of the documents in a collection to one or more smaller groups called clusters. Based on an examination of their words, these clusters should contain similar documents. The initial collection is a single cluster. After processing, the documents are distributed among a number of clusters, where ideally each document is very similar to the other documents in its cluster and much less similar to documents in other clusters.

There are some algorithms can be used to do document clustering. In hierarchical clustering methods, a distance measure is used to build a tree of cluster. When it starts from individual elements and ends with a single cluster, it's called agglomerative. Conversely, it's called divisive when it starts from a complete collection and ends with single objects. Single linkage, complete linkage, and group average are agglomerative clustering which are differentiated by their definition of similarity between clusters. Single linkage defines similarity between clusters based on their most similar pair of objects, whereas complete linkage will do the same task based on their least similar pair of objects, and group average will be based on the average of the similarities.

[1] *Gloria Virginia, S.Kom., MAI., Dosen Teknik Informatika, Fakultas Teknik, Universitas Kristen Duta Wacana*

Another clustering method is partitioning methods. It will divide a set of objects to specific number of cluster. The *k*-means algorithm is a popular method of partitioning methods which can be regarded as a hard clustering method, where each document is uniquely assigned to a single cluster. When a document might belong to different clusters, it can be regarded as using a soft (or fuzzy) clustering method. The fuzzy *c*-means is an instance of it.

### b. Language Modeling for Text Retrieval

There are two basic probabilistic retrieval models [1]. The first model is a generative model of documents from queries, which uses the classical probabilistic approach (Robertson and Sparck Jones, 1976). It's supposed that a document is generated from a query using a binary latent variable that indicates whether or not the document is relevant to the query.

The second model is a generative model of queries from documents. It's supposed that a query is generated from a document, where a language model is estimated for each document. The method of using document language models to assign likelihood scores to queries has come to be known as the *language modeling approach* [12].

A *statistical language model* is a probability distribution over all possible sentences or other linguistic units in a language [14].

### c. Latent Semantic Indexing

Essentially, every word is polysemous, which means has multiple meanings. But at the other hand, there are many ways to express a given concept by a word, which makes a word has synonym with other word. This fact of word brings a problem in information retrieval, while a query expressed by a word is literally matched by words in documents.

This problem is tried to be overcome by *latent semantic indexing* (LSI). It uses statistically derived conceptual indices instead of individual words for retrieval. The key idea in LSI is to map high-dimensional count vectors, such as term-frequency (tf) vectors arising in the vector space representation of text documents [13], to a lower dimensional representation in a so-called latent semantic space. The ultimate goal is to represent semantic relations between words and/or documents in terms of their proximity in the semantic space [10].

### d. Vector Space Model

The most popular family of information retrieval techniques is based on the vector-space model (VSM) for documents [13]. In the VSM, each document is represented by a term vector with (transformed) frequency counts for term occurrences as components. The two most important ingredients of the VSM are: a similarity measure and a term weighting scheme to re-weight the influence of different terms [10]. A successfully applied weighting scheme is the TFIDF (term frequency inverse document frequency). While cosine function is the similarity measure which usually used. It calculates the cosine of the angle between document and query vector.

## 3. Explanation

Document clustering has been used in experimental IR system for decades. It was initially proposed as a means for improving efficiency and also as a way to categorize or classify documents [3]. There is an underlying hypothesis in document clustering, called Cluster Hypothesis. It's stated as follows: closely associated documents tend to be relevant to the same requests [4]. Based on this hypothesis, combined with the use of language model approach, latent semantic indexing, and vector space model, we can overcome the problem of lexical matching in text retrieval, by means of clustering-based retrieval using query-likelihood model.

The approach to cluster-based retrieval is to use cluster as a form of document smoothing [3]. Previous studies have suggested that by grouping documents into clusters, differences between representations of individual documents are, in effect, smoothed out. Here, I'll use the k-*means* algorithm for clustering. Below is the algorithm [5]:

1. Distribute all documents among the *k* bins, randomly.
2. Compute the mean vector for each bin.
3. Compare the vector of each document to the bin means and note the mean vector that is most similar.
4. Move all documents to their most similar bins.
5. If no document has been moved to a new bin, then stop; else go to step 2.

The general idea of query-likelihood model is to build a language model *D* for each document in the collection and rank the documents according to how likely the query *Q* could have been generated from each of these documents models. The most common approach assumes that the query can be treated as a sequence of independence terms, and thus query probability can be represented as a product of the individual term probabilities [15]. We take similar approach for cluster-based retrieval by building language models for cluster then retrieve/rank cluster based on the likelihood of generating the query [3]. Documents in the same cluster are combined and treated as if it were a big document. Below are the equation proposed by Liu and Croft:

$$P(Q|Cluster)=\prod_{i=1}^{m}P(q_i|Cluster) \quad (1)$$

$$P(w|D)=\lambda P_{ML}(w|D)+(1-\lambda)P(w|Cluster)$$
$$=\lambda P_{ML}(w|D)+(1-\lambda)[\beta P_{ML}(w|Cluster)+(1-\beta)P_{ML}(w|Coll)] \quad (2)$$

$$P_{ML}(w|D)=\frac{tf(w,D)}{\sum_{w'\in D}tf(w',D)} \qquad P_{ML}(w|Cluster)=\frac{tf(w,Cluster)}{\sum_{w'\in Cluster}tf(w',Cluster)} \qquad P_{ML}(w|Coll)=\frac{tf(w,Coll)}{\sum_{w'\in V}tf(w',Coll)}$$

Where $q_i$ is the *i*th term in the query, $P(q_i|Cluster)$ is specified by the cluster language model, $P_{ML}(w|D)$ is the maximum likelihood estimate of word *w* in the document, $P_{ML}(w|Cluster)$ Is the maximum likelihood estimate of word *w* in the cluster, $P_{ML}(w|Coll)$ is the maximum likelihood estimate of word *w* in the entire collection, $tf(w,D)$ is the number of times *w* occurs in the document *D*, $tf(w,Cluster)$ is the number of times *w* occurs in the cluster, *V* is the vocabulary, and $tf(w,Coll)$ is the number of times *w* occurs

In the entire collection. Both $\lambda$ and $\beta$ are general symbols for smoothing, and they take different forms when different smoothing methods are applied.

From equation (2) we can see that the cluster model is first smoothed with the collection model, and the document model is then smoothed using the smoothed cluster model. Both of the smooth process is done at once.

The equation proposed by Liu and Croft are similar with the equation proposed by Hofmann [10], which is closely related to the LSI, that's *Probabilistic Latent Semantic Analysis* (PSA). The starting point of PSA is a statistical model which has been called the *aspect model* [11]. To quote from [11]:

> "..the *aspect model* assumes that every *occurrence* of a word in a document is associated with a unique state $z_k$ of the latent class variable. This does by no means exclude that different word occurrences within the same document or occurrences of the same word within different documents can be "explained" by different aspects. However, since latent class variables associated with occurrences in the same document share their prior probabilities $P(z_k, d_i)$ [denotes a document specific probability distribution over the latent variable space], observation within a document get effectively coupled. By symmetry this also holds for different occurrences of the same word. As a result of this coupling, the probabilities $P(z_k, d_i)$ and $P(z_k, w_j)$ [denotes the class-conditional probability of a specific word conditioned on the unobserved class variable $z_k$ ] tend to be "sparse",..."

After having language models for query and document, we measure the similarity between them using cosine function. But here, the original vector space representation of documents is replaced by the language models.

$$sim(d_i, q) = \frac{\vec{d_i} \cdot \vec{q}}{\left|\vec{d_i}\right|\left|\vec{q}\right|} = \frac{\sum_j P(w_j|D) \cdot P(Q|Cluster)}{\sqrt{\sum_j P(w_j|D)^2}\sqrt{\sum_j P(Q|Cluster)^2}} \qquad (3)$$

Based on the result of this similarity calculation, we can form a ranked list of documents by putting documents from the first retrieved cluster at the top followed by those from the second retrieved cluster, and so on.

## 4. Summary

To summarize, here is the algorithm of clustering-based retrieval using query-likelihood model:

a. Organize documents into clusters using k-*means* algorithm.
b. Build the language models for clusters and query.
c. Calculate the similarity between language models of clusters and query using cosine measure.
d. Rank the result documents by combining the documents in all clusters, begun from the most similar cluster.

## 5. References

John Lafferty and Chengxiang Zhai; **Probabilistic IR Models Based on Document and Query Generation**; Proceedings of the Workshop on Language Modeling and Information Retrieval; Carnegie Mellon University, 2001.

John Lafferty and ChengXiang Zhai; **Probabilistic Relevance Models Based on Document and Query Generation**; In *Language Modeling and Information Retrieval*; Kluwer International Series on Information Retrieval; Vol. 13; 2003

Xiaoyong Liu and W Bruce Croft; **Cluster-Based Retrieval Using Language Models**; 2004 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'04); Sheffield, United Kingdom; 2004.

C.J. Van Rijsbergen; **Information Retrieval**; 2nd edition; London: Butterworths; 1979.

Sholom M. Weiss, Nitin Indurkhya, Tong Zhang, Fred J. Damerau; **Text Mining: Predictive Methods For Analyzing Unstructured Information**; New York: Springer Science+Business Media, Inc.; 2005.

L. Kaufman; **Cluster Analysis**; Université de Liège; 1980.

Jay M. Ponte and W. Bruce Croft; **A Language Modeling Model Approach to Information Retrieval**; 1998 ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98); Melbourne, Australia; 1998.

Michael W. Berry and Murray Browne; **Understanding Search Engines: Mathematical Modeling and Text Retrieval**; Philadelphia: Society for Industrial and Applied Mathematics, University City Science Center; 1999.

M-F. Moens; **Text Clustering**; In *course document of Text Based Information Retireval course*; Katholieke Universiteit Leuven; 2005.

Thomas Hofmann; **Unsupervised Learning by Probabilistic Latent Semantic Analysis**; In *Machine Learning*; Vol. 42, 177-196; 2001.

Thomas Hofmann, J. Puzicha, and M. I. Jordan; **Unsupervised Learning from Dyadic Data**; In *Advances in Neural Information Processing Systems*; Vol. 11; MIT Press; 1999.

Http://www.lemurproject.org/3.1/background.html

D. Miller, T. Leek, and Schwartz, R.; **A Hidden Markov Model Information Retrieval System** ; In *SIGIR*; pp. 214-221; 1999.

G. Salton and M. J. McGill; **Introduction to Modern Information Retrieval**; New York: McGraw-Hill; 1983.

R. Rosenfeld; **Two Decades of Statistical Language Modeling: Where Do We Go From Here?**; In *Proceedings of the* IEEE; Vol. 88(8); 2000.