TANGUDU NARESH, G.RAMESH NAIDU, S.VISHNU MURTY / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
www.ijera.com Vol. 2, Issue 1,Jan-Feb 2012, pp.1044-1049

Learning Subject Areas by Using Unsupervised Observation of Most Informative Terms in Text Databases

TANGUDU NARESH*, G.RAMESH NAIDU**, S.VISHNU MURTY***

*(M.Tech Scholar,CSE, Pydah engineering college, Vishakhapatnam,AP,India &Asst Professor. Dept.of.MCA, Aditya Institute of Technology And Management, Tekkali,AP,India-532201)
**(Assoc.Professor&HOD,Dept.of.CSE, Pydah Educational Institutions,Vishakhapatnam,AP,India)
***(Sr.Asst.Professor ,Dept.of.MCA, Aditya Institute of Technology And Management, Tekkali,AP,India-532201)

Abstract:
Much work has been done on automatic topic detection using learning by example techniques [1] but they are confined to assignment of terms onto a predetermined classes using some algorithms which are trained on some manually classified documents and most of them identify only text boundaries. But in the present we do not use any prior information for subject area identification. Here we scan most informative/prominent words from the corpus [2] and they are grouped using learning by observation techniques [3] and some of the well known distance measures [4,5] for subject area identification. We may use similarity measures like total divergence to the average [6], Bhattacharya co-efficient [7] for term similarities.

Keywords: ***Similarity measures, (k-means) partitional clustering, text clustering, corpus, stop list, word bag***

1.Introduction

we consider the problem of finding the set of most prominent topics in a collection of documents without using any prior knowledge or fixed list of topics i.e unsupervised learning .We do not rely on the training set or other forms of external knowledge but we have to get by with the information contained in the collection itself. We found that this works fairly well in an evaluation with Wikipedia articles ,Many web sites (such as YouTube, The New York Times, eBay, and Google Base) function on top of large databases and offer a variety of services in which we com-pared human defined topic categories with subject Area clusters.

Unsupervised learning of informative terms requires a similarity measure or distance measure between terms(words).In this paper we consider distance measures between informative words that are based on the statistical distribution of words in a corpus of texts. . The focus of is to find a measure that yields good clustering results.

The organization of this paper is as follows. In section 2 we discuss background and extraction of informative words. In section 3 we discuss distance measures and introduce different probability densities needed to define them. Section 4 describes the brief note on clustering technique used for subject area categorization. Section 5 we present an evaluation of topic detection on a Wikipedia corpus using clustering of keywords with different distance measures section 6 discuss about future enhancement of proposed system.

2. BackGround

Much work has been done on automatic text categorization but most of this work is connected with assignment of texts into a small set of given categories. In many cases some form of machine learning is used to train an algorithm on a set of manually categorized documents i.e supervised learning.

Moreover they focus on determining the boundaries and the topic of short paragraphs while we try to find the predominant overall topic of a whole text.

But we consider the problem of finding the set of most prominent topics in a collection of documents without using any prior knowledge or fixed list of topics i.e unsupervised learning .We do not rely on the training set or other forms of external knowledge but we have to get by with the information contained in the collection itself.

Similar the work presented in this paper by Scott Deerwester, Susan Dumais, George Furnas, and Richard Harshman try to identify the delineation of subject area by using some stastical distance measures like Kullback–Leibler divergence [15,16,17], Bhattacharyya coefficient[7]. These measures often similar bur slightly differ from total divergence to the average.

Here above and our works concerned related to LSI [8] and PLSI [9]. The input data for both is word weight and conjugation data of informative terms. LSI assumes that words that are close in meaning will occur

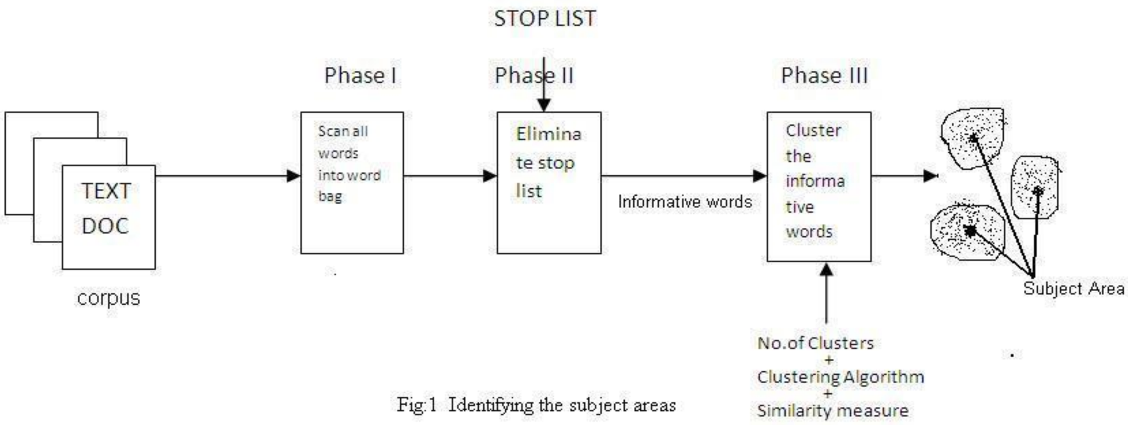

Fig:1 Identifying the subject areas

close together in text. A matrix containing word counts per paragraph (rows represent unique words and columns represent each paragraph) is constructed from a large piece of text and a mathematical technique called singular value decomposition (SVD)[10,11] is used to reduce the number of columns while preserving the similarity structure among rows. Words are then compared by taking the cosine of the angle between the two vectors formed by any two rows. Values close to 1 represent very similar words while values close to 0 represent very dissimilar words where as PLSI evolved from latent semantic indexing, adding a sounder probabilistic model i.e probabilistic latent semantic indexing is based on a mixture decomposition derived from a latent class model[12].sometimes often they use In statistics, latent Dirichlet allocation (LDA)[13] for the above work. In our work the term clusters are similarly based on co-occurrence of data. This is done by comparing cluster densities of co-occurring of terms. The representative of the cluster is average density of co occurrence densities.

3. Grouping informative codewords

In this we will group the informative terms based on some criteria for similarity i.e distance measure on terms i.e in turn define on as similarity between probability densities associated with terms by counting occurrences in documents. The work depicted by following fig: 1 will be done in 3 phases.

Phase-01: All the text documents are scanned in to Word bag input to this phase is set of text documents i.e The corpus

Phase 02: All irrelevant words i.e stop list is eliminated from the word bag . Input to this phase is stop list and the scanned all words

Phase-03: The remaining terms i.e. keywords are clustered using distance measures like cosine similarity. The input to this phase is number of desired clusters and the keywords. The clustering algorithm used is K-means, a partition based clustering algorithm

3.1 Spotting informative word

A technique Yahoo Term Extraction[24], Wikipedia Term Extraction[25],The Web can be used to identify terms that tend to co-occur frequently to identify the important terms in a text document using Wikipedia, Amazon Mechanical Turk service .but This not main focus of the paper rather we consider the nouns, verbs and proper names from corpus . The conflicting terms or irrelevant words i.e. stop list is eliminated.

3.2 Probability densities

We simplify a document to a bag of words, terms or keywords, in the following always called terms. We consider a collection of n term occurrences W. Each term occurrence is an instance of exactly one term t in T = $\{t_1, \dots t_m\}$, and can be found in exactly one source document d in a collection C = $\{d_1, \dots d_M\}$. Let n(d, t) be the number of occurrences of term t in d, n(t) = $\sum_d n(d,t)$ be the number of occurrences of term t, and N(d) =$\sum_t n(d,t)$ the number of term occurrences in d.

Now let us consider the algorithm for term frequency for informative word in corpus.

Input: Original database (list of corpus), stop list
Output: Annotated database that containing the informative words.
For each document d in corpus do
Extract all terms from d
/* Compute term frequencies */
For each term t in d do

If t is not in stop list
Freq(t) = Freq(t) + 1
End if
End for each
End for each

We consider the natural probability distributions **Q** on C × T , a distribution Q on C and q on T that measure the probability to randomly select an occurrence of a term, from a source document or both

Q(d, t) = n(d, t)/n on C × T Q(d) = N(d)/n on C (1)
q(t) = n(t)/n on T (2)

These distributions are the baseline probability distributions for everything that we will do in the remainder. In addition we have two important conditional probabilities

$Q(d|t) = Q_t(d) = n(d, t)/n(t)$ on C (3)

$q(t|d) = q_d(t) = n(d, t)/N(d)$ on T (4)

The suggestive notation Q(d|t) is used for the source distribution of t as it is the probability that a randomly selected occurrence of term t has source d. Similarly, q(t|d), the term distribution of d is the probability that a randomly selected term occurrence from document d is an instance of term t. Various other probability distributions on C × T , C and T that we will consider will be denoted by **P**, P , p respectively, dressed with various sub and superscripts.
Distributions of Co-occurring Terms The setup in the previous section allows us to set up a Markov chain on the set of documents and terms which will allow us to propagate probability distributions from terms to document and vice versa. Consider a Markov chain on T× C having transitions C → T with transition probabilities Q(d|t) and transitions T → C with transition probabilities q(t|d) only.

Given a term distribution p(t) we compute the one step Markov chain evolution. This gives us a document distribution $P_p(d)$, the probability to find a term occurrence in a particular document given that the term distribution of the occurrences is p

$$P_p(d) = \sum Q(d|t)p(t). \quad (5)$$

Likewise given a document distribution P (d), the one step Markov chain evolution is the term distribution

$$P_p(t) = \sum q(t|d)p(d) \quad (6)$$

Since P (d) gives the probability to find a term occurrence in document d, p_P is the P-weighted average of the term distributions in the documents. Combining these, i.e. running the Markov chain twice, every term distribution gives rise to a new distribution.
The distribution of co-occurring terms $\bar{p}z$ is

$$\overline{p_z}(t) = \sum_{d,t'} q(t|d)Q(d|t')p_z(t') = \sum_{d} q(t|d)Q(d|z)$$

(6)

3.3. Distance Measures

An effective way to define "similarity" between two elements is through a metric d(i, j) between the elements i, j satisfying the usual axioms of nonnegative, identity of in-discernable and triangle inequality. Two elements are more similar if they are closer. For this purpose any monotone in-creasing function of a metric will suffice and we will call such a function a distance function.

For clustering we use a hierarchical top-down method that requires that in each step the center of each cluster is computed. Thus our choice of distance function is restricted to distances defined on a space allowing us to compute a center and distances between keywords and this center. In particular we cannot use popular similarity measures like the Jaccard coefficient [14].

In the following we will compare results with four different distance functions for keywords t and s: (a) the cosine similarity of the document distribution Q_t and Q_s considered as vectors on the document space, (b) the cosine similarity of the vectors of tf.idf values[23] of keywords, (c) the Total divergence to the average between the document distributions Q_t and Q_s and (d) the Total divergence to the average between the term distributions, $\overline{p_t}$ and $\overline{p_s}$.

The cosine similarity of two terms t and s is defined as

$$\cos sim_{tf}(t,s) = \frac{\sum_{dec} Q_t(d) Q_s(d)}{\sqrt{(\sum_{dec} Q_t^2(d)(\sum_{dec} Q_s^2(d))}} \quad (7)$$

i.e cosine simlairty[4,5] The cosine of two vectors can be$_t$ easily derived by using the Euclidean Dot Product formula: Given two vectors of attributes, *A* and *B*, the cosine similarity, *θ*, is represented using a dot product and magnitude as

$$\text{Similarity} = \cos(\theta) = \frac{A, B}{\|A\|, \|B\|} = \frac{\sum_{i=1}^{n} A_i X B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2}\, X \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

(8)

Since the arcos of this similarity function is a proper metric, (1 − cos)(arcos(cos sim(t, s))) = 1 − cos sim(t, s) is a distance function.

The Total divergence to the average or information radius [5] between two distributions p and q is defined as

$$TDA(P\|Q) = \frac{1}{2}D(P\|M) + \frac{1}{2}D(Q\|M)$$

(9)

$$\text{where } M = \frac{1}{2}(P + Q)$$

D(P||M) is the relative entropy or Kullback-Leibler divergence[15,16,17] between p and m is define as

$$D_{KL}(P\|M) = \sum_i P(i) \log \frac{P(i)}{M(i)}$$

(10)

Consider the set $M_{+}1(A)$of probability distributions where A is a set provided with some σ-algebra [18]of measurable subsets. If A is countable, a more general definition, allowing for the comparison of more than two distributions, is:

$$TDA(P_1, P_2, \ldots\ldots\ldots, P_n) = H\left(\sum_{i=1}^{n} \pi_i P_i\right) - \sum_{i=0}^{n} \pi_i H(P_i)$$

(11)

Where $\pi_1,\pi_2,\pi_3,\pi_4\ldots\ldots\ldots\pi_n$ are the weights for the probability distributions and *H*(*P*) is the Shannon entropy for distribution *P*. For the two-distribution case described above,

$$P_1 = P,\ P_2 = Q,\ \pi_1 = \pi_2 = \frac{1}{2} \quad (12)$$

$$D(p\|q) = \sum_{i=1}^{n} p_i \log\left(\frac{p_i}{q_i}\right).$$

Since the square root of the Total divergence to the average [6] is a proper metric [19], we have two distances

$$\text{TDA sim}_{doc}(t, s) = \text{TDA}(Q_t, Q_s) \quad (13)$$

And

$$\text{TDA sim}_{term}(t, s) = \text{TDA}(\overline{p_s}, \overline{p_t}) \quad (14)$$

3.4. Clustering Method

Bisecting K-Means Algorithm

- Pick a cluster to split (split the largest
- Find 2 sub-clusters using the basic K-means algorithm
- Repeat step 2, the bisecting step, for n times and take the split that produces the clustering with the highest overall similarity
- Repeat steps 1, 2 and 3 until the desired number of clusters is reached

We have used the induced bisecting k-means clustering algorithm [20] as described by , which is based on the standard bisecting k-means algorithm. But initially we find only two clusters by selecting two elements that have largest distance which we use as seeds for the two clusters. All other terms are assigned to one of the clusters based on closeness. Centers of the clusters are computed. Now we have found two clusters. Now if the diameter of cluster is larger than a specified threshold value the whole procedure is recursively applied on that cluster.

3.5 Experimental Results

Several artificially generated text documents and real world datasets were used to experimentally demonstrate that the identification of informative word by learning by observation [3] is able to work fairly well using different similarity measures.

To evaluate the implemented topic detection methods, we have compared the results with topics known to be present in the collection. We benchmarked against the 8 selected Wikipedia topics of the collection. Of course, it is conceivable that the collection has more topics that automatic methods might recognize as well. To define a reference clustering, we have clustered the 160 selected keywords into a set of 9 categories C*= $\{c_0^*, c_1^* \ldots \ldots c_8^*\}$, one for each Wikipedia category and a rest cluster c_0^*, using the following method. For each of the 8 Wikipedia categories c_i^* we compute the distribution $q_{c_i^*}$ of words in the documents belonging to c_i^* and we let $q_{c_0^*}$ = q. We assign a term z to cluster c*

$$\text{if } c^* = \text{argmin}_{c^* \in C} D(q_{c^*} \| \overline{p_z}) \quad (15)$$

We now compare with the set of clusters C of keywords found using the algorithm in section 3.4, different distance measures and different diameters. For each cluster c Є C and cluster c*Є C*we define a recall Measure

$$rec(c, c^*) = |c \cap c^*| / |c^*| \quad (16)$$

Precision measure

$$prec(c, c^*) = |c \cap c^*| / |c^*| \quad (17)$$

and an F value

$$F(c,c^*) = \frac{rec(c,c^*)prec(c,c^*)}{\frac{1}{2}(rec(c,c^*) + prec(c,c^*))} \quad (18)$$

Let $F(c^*) = \max_{c \in C} F(c, c^*)$ be the F-value of the best fitting found cluster and finally define the overall F-value.

$$F = \sum_{c^* \in C^*} \frac{|C^*|}{\sum_{c^* \in C^*} |C^*|} F(C^*) \quad (19)$$

The overall F-values for clustering with the different similarities are given in Figure.2.

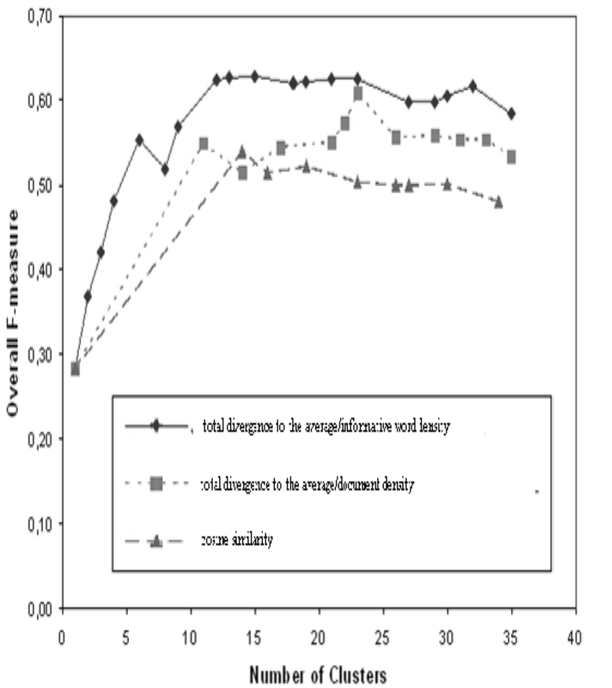

Figure.2

4. Future enhancements

In above 3.4 we partition a dataset into a fixed number of clusters supplied by the user manually. The estimation of number of clusters for partitioning the dataset (list of informative words) is difficult in the case of large text databases, sometimes which lead to inefficient data distribution or majority outliers. Hence, in future enhancement of this paper we propose a noble method using rotation estimation also called Cross-Validation [21] which identifies a suitable number of clusters in a given unlabeled dataset without using prior knowledge about the number of clusters or we use any model based unsupervised learning [22].

5. Conclusion

The pragmatic results suggest that the spotting the informative word by using un supervised learning works fairly well on corpus by using different similarity measures. But the computation of informative word density is expensive because of conjugation of informative words.

References

[1] Jiawei Han and Michelin Kamber, Data Mining: Concepts and Techniques, 2nd ed.The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor
Morgan Kaufmann Publishers, March 2006. ISBN 1-55860-901-6 chapter 8; page 500

[2] Sinclair, J. 'The automatic analysis of corpora', in Svartvik, J. (ed.) *Directions in Corpus Linguistics (Proceedings of Nobel Symposium 82)*. Berlin: Mouton de Gruyter. 1992.

[3] Jiawei Han and Michelin Kamber, Data Mining: Concepts and Techniques, 2nd ed.The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor
Morgan Kaufmann Publishers , March 2006. ISBN 1-55860-901-6 chapter 8; page 500

[4] Similarity Measures for Text Document Clustering, Anna Huang Department of Computer Science,The University of Waikato, Hamilton, New Zealand, *NZCSRSC 2008*, April 2008

[5] http://en.wikipedia.org/wiki/Category :Statistical_distance_measures

[6] B. Fuglede and F. Topsoe. Jensen-shannon divergence and hilbert space embedding. In Proc. of the Internat. Symposium on Information Theory, 2004, pages 31–, 2004.

[7] A. Bhattacharyya. On a Measure of Divergence between Two Statistical Populations De_ned by their Probability Distributions. Bulletin of Calcutta Mathematical Society, 35:99:110,1943.

[8] Susan T. Dumais (2005). "Latent Semantic Analysis". *Annual Review of Information Science and Technology* 38: 188. doi:10.1002/aris.1440380105

[9] Thomas Hofmann (1999). "Probabilistic Latent Semantic Analysis" (PDF). *Uncertainty in Artificial Intelligence*. http://www.cs.brown.edu/people/th/papers/Hofmann-UAI99.pdf.

[10] Trefethen, Lloyd N.; Bau III, David (1997). *Numerical linear algebra*. Philadelphia: Society for Industrial and Applied Mathematics. ISBN 978-0-89871-361-9.

[11] Wall, Michael E., Andreas Rechtsteiner, Luis M. Rocha (2003). "Singular value decomposition and principal component analysis". In D.P. Berrar, W. Dubitzky, M. Granzow. *A Practical Approach to*

Microarray Data Analysis. Norwell, MA: Kluwer. pp. 91–109. http://public.lanl.gov/mewall/kluwer2002.html.

[12] Linda M. Collins, Stephanie T. Lanza (2010). *Latent class and latent transition analysis for the social, behavioral, and health sciences*. New York: Wiley. ISBN 9780470228395

[13] Blei, David M.; Ng, Andrew Y.; Jordan, Michael I (January 2003). Lafferty, John. ed. "Latent Dirichlet allocation". *Journal of Machine Learning Research* 3 (4–5): *pp.* 993–1022. doi:10.1162/jmlr.2003.3.4-5.993. http://jmlr.csail.mit.edu/papers/v3/blei03a.html.

[14] Tan, Pang-Ning; Steinbach, Michael; Kumar, Vipin (2005), *Introduction to Data Mining*, ISBN 0-321-32136-7

[15] Kullback, S.; Leibler, R.A. (1951). "On Information and Sufficiency". *Annals of Mathematical Statistics* 22 (1): 79–86. doi:10.1214/aoms/1177729694. MR39968.

[16] S. Kullback (1959) *Information theory and statistics* (John Wiley and Sons, NY).

[17] Kullback, S.; Burnham, K. P.; Laubscher, N. F.; Dallal, G. E.; Wilkinson, L.; Morrison, D. F.; Loyer, M. W.; Eisenberg, B. et al. (1987). "Letter to the Editor: The Kullback–Leibler distance". *The American Statistician* 41 (4): 340–341. JSTOR 2684769.

[18] Rudin, Walter (1987). *Real & Complex Analysis*. McGraw-Hill. ISBN 0-07-054234-1.

[19] B. Fuglede and F. Topsoe. Jensen-shannon divergence and hilbert space embedding. In Proc. of the Internat. Symposium on Information Theory, 2004, pages 31–, 2004

[20] F. Archetti, P. Campanelli, E. Fersini, and E. Messina. A hierarchical document clustering environment based on the induced bisecting k-means. In H. L. Larsen, G. Pasi, D. O. Arroyo, T. Andreasen, and H. Christiansen, editors, FQAS, volume 4027 of Lecture Notes in Computer Science, pages 257–269. Springer, 2006.

[21] Kohavi, Ron (1995). "A study of cross-validation and bootstrap for accuracy estimation and model selection". *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* 2 (12): 1137–1143. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.529.(Morgan Kaufmann, San Mateo)

[22] G.S.N. Murthy et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (2) , 2011, 607-610 Automation of Data Clusters based on Layered HMM.

[23]. http://en.wikipedia.org/wiki/Tf.idf

[24] P. Wu, Y. Sismanis, and B. Reinwald, “Towards keyword-driven analytical processing,” in Proceedings of the 2007 ACM SIGMOD nternational Conference on Management of Data (SIGMOD 2007), 2007.

[25] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey,“Scatter/Gather: A cluster-based approach to browsing large document collections,” in Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,SIGIR’92, 1992, pp. 318–329.