Topic Detection by Clustering Keywords

Christian Wartena and Rogier Brussee
Telematica Instituut, P.O. Box 589, 7500 AN Enschede, The Netherlands
{Christian.Wartena,Rogier.Brussee}@telin.nl

Abstract

We consider topic detection without any prior knowledge of category structure or possible categories. Keywords are extracted and clustered based on different similarity measures using the induced k-bisecting clustering algorithm. Evaluation on Wikipedia articles shows that clusters of keywords correlate strongly with the Wikipedia categories of the articles. In addition, we find that a distance measure based on the Jensen-Shannon divergence of probability distributions outperforms the cosine similarity. In particular, a newly proposed term distribution taking co-occurrence of terms into account gives best results.

1. Introduction

In this paper we consider the problem of finding the set of most prominent topics in a collection of documents. Since we will not start with a given list of topics, we treat the problem of identifying and characterizing a topic as an integral part of the task. As a consequence, we cannot rely on a training set or other forms of external knowledge, but have to get by with the information contained in the collection itself. The approach we will follow consists of two steps. First we extract a list of the most informative keywords. Subsequently we try to identify clusters of keywords for which we will define a center, which we take as the representation of a topic. We found that this works fairly well in an evaluation with Wikipedia articles in which we compared human defined topic categories with keyword clusters.

Clustering of (key)words requires a similarity or distance measure between words. In this paper we consider distance measures between 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 related work and the related problem of keyword extraction. In section 3 we discuss distance functions and introduce different probability distributions needed to define them. We end the section with a brief description of the clustering algorithm used for evaluation. Finally, in section 4 we present an evaluation of topic detection on a Wikipedia corpus using clustering of keywords with different distance measures.

2. Related Work

Much work has been done on automatic text categorization. Most of this work is concerned with the assignment of texts onto 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. A lot of work has been done on clustering of texts (See e.g. [14] and [12]) which has already found practical applications like the clustering of search results (see e.g. `http://clusty.com/`).

The topic of the clusters remains usually implicit in these approaches, though it would of course be possible to apply any keyword extraction algorithm to the resulting clusters in order to find characteristic terms.

Like the work presented in this paper, Li and Yamanishi ([10, 9]) try to find characterizations of topics directly by clustering keywords using a statistical similarity measure. While very similar in spirit, their similarity measure is slightly different from the Jensen-Shannon based similarity measure we use. 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.

Both our and their work are conceptually related to latent semantic analysis (LSA) [3, 7] and even more so to probabilistic latent semantic analysis (PLSA) [5, 6] and related work by [15]. The input for both our methods and (P)LSA is the word occurrence and co-occurrence data of terms. Latent semantic analysis now naturally leads to weighted sums of terms, with the crucial difference that in the PLSA case only non negative weights with sum 1 are allowed which have a natural interpretation as conditional probabilities. In the latent semantic analysis approach the analogue of clustering is finding (latent) conditional probability distributions such that we can decompose the observed word distributions into a few of these latent distributions and a small "noisy" remainder. In this decomposition words with strong mutual co-occurrence tend to have the same main latent components. In our work the word clusters are similarly based on co-occurrence data. This is achieved by comparing and clustering distributions of co-occurring terms. Thus, the center of a cluster is the average distribution of the co-occurrence distributions, and is in this sense comparable to a latent component.

3. Clustering Keywords

By clustering we will understand grouping terms, documents or other items together based on some criterion for similarity. We will always define similarity using a distance function on terms, which is in turn defined as a distance between distributions associated to (key)words by counting (co)occurrences in documents.

3.1. Keyword Extraction

We will represent a topic by a cluster of keywords. We therefore need a set of keywords for the corpus under consideration. Note that finding a set of keywords for a corpus is not the same problem as assigning keywords to a text from a list of keywords, nor that of finding the most characteristic terms for a given subset of the corpus. The problem of finding a good set of keywords is similar to that of determining term weights for indexing documents, and not the main focus of this paper. For our purposes usable results can be obtained by selecting the most frequent nouns, verbs (without auxiliaries) and proper names and filtering out words with little discriminative power (see section 4).

3.2. Distributions

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 $\mathcal{W}$. Each term occurrence is an instance of exactly one term t in $\mathcal{T} = \{t_1, \dots t_m\}$, and can be found in exactly one source document d in a collection $\mathcal{C} = \{d_1, \dots d_M\}$. 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.

We consider the natural probability distributions $\mathbf{Q}$ on $\mathcal{C} \times \mathcal{T}$, a distribution Q on $\mathcal{C}$ and q on $\mathcal{T}$ that measure the probability to randomly select an occurrence of a term, from a source document or both

$$\begin{aligned}\mathbf{Q}(d,t) &= n(d,t)/n \text{ on } \mathcal{C} \times \mathcal{T} \\ Q(d) &= N(d)/n \text{ on } \mathcal{C} \\ q(t) &= n(t)/n \text{ on } \mathcal{T}\end{aligned}$$

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

$$\begin{aligned}Q(d|t) &= Q_t(d) = n(d,t)/n(t) \text{ on } \mathcal{C} \\ q(t|d) &= q_d(t) = n(d,t)/N(d) \text{ on } \mathcal{T}\end{aligned}$$

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 $\mathcal{C} \times \mathcal{T}$, $\mathcal{C}$ and $\mathcal{T}$ that we will consider will be denoted by $\mathbf{P}$, 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 $\mathcal{T} \cup \mathcal{C}$ having transitions $\mathcal{C} \to \mathcal{T}$ with transition probabilities $Q(d|t)$ and transitions $\mathcal{T} \to \mathcal{C}$ 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

$$P_p(d) = \sum_t Q(d|t)p(t).$$

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

$$p_P(t) = \sum_d q(t|d)P(d).$$

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

$$\bar{p}(t) = p_{P_p}(t) = \sum_d q(t|d)P_p(d) = \sum_{t',d} q(t|d)Q(d|t')p(t')$$

In particular starting from the degenerate "known to be z" term distribution

$$p_z(t) = p(t|z) = \begin{cases} 1 & \text{if } t = z\ , \\ 0 & \text{otherwise.} \end{cases} ,$$

the *distribution of co-occurring terms* $\bar{p}_z$ then is

$$\bar{p}_z(t) = \sum_{d,t'} q(t|d)Q(d|t')p_z(t') = \sum_d q(t|d)Q(d|z).$$

This distribution is the weighted average of the term distributions of documents containing z where the weight is the probability $Q(d|z)$ that a term occurrence in d is an occurrence of z.

Note that the probability measure $\bar{p}_z$ is very similar to the setup in [10, section 3]. The difference is that we keep track of the density of a keyword in a document rather than just the mere occurrence or non occurrence of a keyword in a document. This difference is particularly relevant for long documents where a word may occur with very low density, yet because it contains many words have a significant contribution to the mean word distribution. Unfortunately $\bar{p}_z$ is expensive to compute.

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 nonnegativity, identity of indiscernables and triangle inequality. Two elements are more similar if they are closer. For this purpose any monotone increasing 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.

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 of keywords, (c) the Jensen-Shannon divergence between the document distributions Q_t and Q_s and (d) the Jensen-Shannon divergence between the term distributions, $\bar{p}_t$ and $\bar{p}_s$.

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

$$\cos\mathrm{sim}_{tf}(t, s) = \frac{\Sigma_{d\in C} Q_t(d)Q_s(d)}{\sqrt{\left(\sum_{d\in C} Q_t^2(d)\right)\left(\sum_{d\in C} Q_s^2(d)\right)}}.$$

Since the $\arccos$ of this similarity function is a proper metric, $(1 - \cos)(\arccos(\cos\mathrm{sim}(t, s))) = 1 - \cos\mathrm{sim}(t, s)$ is a distance function.

The Jensen-Shannon divergence or information radius [11, 4] between two distributions p and q is defined as

$$\mathrm{JSD}(p||q) = \frac{1}{2}D(p||m) + \frac{1}{2}D(q||m)$$

where $m = 1/2(p+q)$ is the mean distribution and $D(p||q)$ is the relative entropy or Kullback-Leibler divergence between p and q which is defined by

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

Since the square root of the Jensen Shannon divergence is a proper metric [4], we have two distances

$$\mathrm{JSD\,sim}_{doc}(t, s) = \mathrm{JSD}(Q_t, Q_s)$$

and

$$\mathrm{JSD\,sim}_{term}(t, s) = \mathrm{JSD}(\bar{p}_s, \bar{p}_t)$$

3.4. Clustering Method

We have used the induced bisecting k-means clustering algorithm as described by [1], which is based on the standard bisecting k-means algorithm, see e.g. [14]. An informal description of the method is as follows. We start by selecting two elements that have the largest distance, which we use as the seeds for two clusters. Next all other items are assigned to the cluster closest to one of the two seeds. After all items have been assigned to a cluster, the centers of both clusters are computed. Here we need a representation of items that naturally allows to define a center which typically is not an item proper but a weighted sum of items. The new centers serve as new seeds for finding two clusters and the process is repeated until the two centers are converged up to some predefined precision. We have now found two clusters. If the diameter of a cluster is larger than a specified threshold value, the whole procedure is applied recursively to that cluster. The algorithm therefore finds a binary tree of clusters.

We also experimented with agglomerative hierarchical clustering, especially the single link algorithm. First results suggested that this approach performed not as well as the k-means algorithm, in line with similar findings by [14] and [12]. Since our focus is to find an optimal distance measure and not to find the best clustering algorithm, we did not investigate the agglomerative methods in more detail.

4. Evaluation

4.1. Implementation

To test and compare the different strategies we have compiled a small corpus of Dutch Wikipedia articles consisting of 758 documents. In the analysis phase, 118099 term occurrences, and 26373 unique terms were found. The articles were taken from 8 Wikipedia categories: spaceflight, painting, architecture, trees, monocots, aviation, pop music, charadriiformes. Categories were selected for subjective similarity, like spaceflight and aviation, and subjective dissimilarity like pop music and monocots. Articles are equally distributed over the categories, but articles in some categories are significantly longer than in others. Moreover, homogeneity and specifity of articles differs significantly between categories.

To determine a set of relevant keywords we have selected the most frequent content words and filtered out a number of overly general terms. The latter was done by requiring that a keyword has to be different enough from the background distribution q, as measured by the Kullback-Leibler divergence. We used a cutoff $D(\bar{p}_t||q) > 1$ bit, that turned out to give decent results.

Before extracting keywords we do some preprocessing using the GATE–framework [2]. The main analysis steps are: lemmatization, multiword lookup and named entity recognition. Lemmatization and tagging is done using the Treetagger [13]. Tagging allows us to distinguish content words from function words. After lemmatization all inflected forms of verbs, nouns and adjectives are mapped onto their lexical form, substantially reducing the variation in the input data. For multiword lookup we used article titles from the Dutch Wikipedia, since it is reasonable to assume that each title represents a single concept. Finally, some simple rules are applied to identify proper names. While some of the components are language dependent, all of the components are available for a number of languages within the GATE–framework.

We selected the 160 most frequent keywords fulfilling this criterion and clustered them with the induced bisecting k-means algorithm from section 3.4 using different distance measures.

4.2. Results

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 $C^* = \{c_0^*, c_1^*, \cdots, c_9^*\}$, 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^* if $c^* = \operatorname{argmin}_{c^* \in C} D(q_{c^*}||\bar{p}_z)$.

Following [8], 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 $c \in C$ and cluster $c^* \in C^*$ we define a recall measure $\text{rec}(c, c^*) = |c \cap c^*|/|c^*|$, a precision measure $\text{prec}(c, c^*) = |c \cap c^*|/|c^*|$ and an F value

$$F(c, c^*) = \frac{\text{rec}(c, c^*)\text{prec}(c, c^*)}{\frac{1}{2}\left(\text{rec}(c, c^*) + \text{prec}(c, c^*)\right)}.$$

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

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

A value of $F = 1$ therefore means that for each Wikipedia and rest category the topic detection algorithm found a corresponding cluster.

Since the clustering algorithm can produce clusters of different size, the quality of the result depends on the only input parameter of the algorithm, the threshold value to split up a cluster. Its quality (in terms of overall F-value) is compared between distance functions by varying the number of clusters as controlled by varying the threshold values. Since there might be several clusters with exactly the same largest distance between two elements, we do not find values for each number of clusters. In particular, for $\cos \text{sim}_{tf}$ and JSD sim_{doc} there are no values that yield a clustering into less than 14 and 11 clusters, resp. In the case of cosine similarity there are in each of these clusters even two keywords with completely orthogonal document distributions. The overall F-values for clustering with the different similarities are given in Figure 1. Cosine similarity with *tf.idf* vectors performed in between direct cosine similarity and the Jensen-Shannon distance with the document distribution

The rest category in the reference categorization is not a real cluster or category of keywords. Moreover, this category is about three times as large as the average size of the other categories. We have therefore tested how well the 8 positively motivated categories are detected, and computed the overall F-values averaging only over these 8 categories. The results are given in Figure 2. Results for the cosine similarity with *tf.idf* vector were slightly better than clustering with JSD sim_{doc}

5. Conclusions

The experimental results suggest that topic identification by clustering a set of keywords works fairly well, using either of the investigated similarity measures. In the present experiment a recently proposed distribution of terms associated with a keyword clearly gives best results, but computation of the distribution is relatively expensive. The reason for this is the fact that co-occurrence of terms is (implicitly) taken into account. The document distribution for terms,

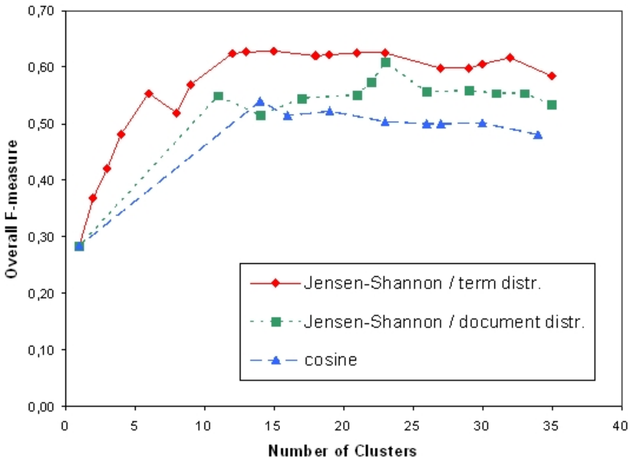

Figure 1. Overall F-Measure for clustering based on 9 categories

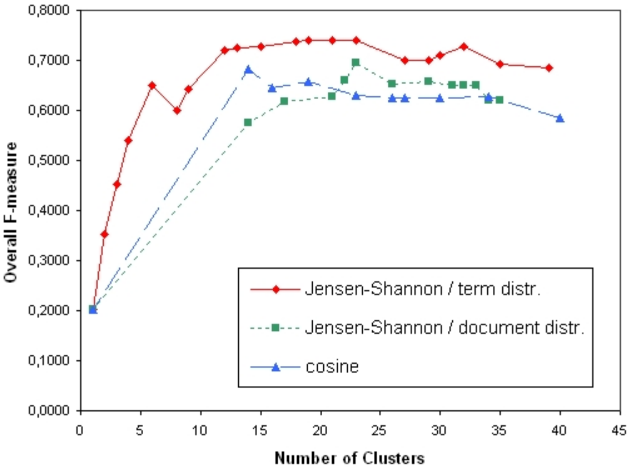

Figure 2. Overall F-Measure for clustering based on 8 categories

that is the base of the other measures, tend to be very sparse vectors, since for a given document most words will not occur at all in that document. In the distribution used for the JSD sim_{doc} this problem alleviated by a kind of 'intelligent smoothing' or spreading values from one term to frequently co-occurring terms.

Acknowledgements

This work is part of the MultimediaN project[1] sponsored by the Dutch government under contract BSIK 03031.

References

[1] 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.

[2] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In *Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, pages 168–175, Philadelphia, July 2002.

[3] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman. Using latent semantic analysis to improve access to textual information. In *CHI '88: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 281–285, New York, NY, USA, 1988. ACM.

[4] 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.

[5] T. Hofmann. Probabilisitic latent semantic analysis. In *UAI99: Uncertainty in artificial intelligence*, 1999.

[6] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, January 2001.

[7] T. Landauer, P. Foltz, and D. Laham. Introduction to latent semantic analysis. *Discourse Processes*, 25:259–284, 1998.

[8] B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *KDD*, pages 16–22, 1999.

[9] H. Li and K. Yamanishi. Text classification using esc-based stochastic decision lists. *Inf. Process. Manage.*, 38(3):343–361, 2002.

[10] H. Li and K. Yamanishi. Topic analysis using a finite mixture model. *Inf. Process. Manage.*, 39(4):521–541, 2003.

[11] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.

[12] S. Meyer zu Eissen and B. Stein. Analysis of clustering algorithms for web-based search. In D. Karagiannis and U. Reimer, editors, *PAKM*, volume 2569 of *Lecture Notes in Computer Science*, pages 168–178. Springer, 2002.

[13] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, Manchester, UK, 1994. unknown.

[14] M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. *In Proceedings of Workshop on Text Mining, 6th ACM SIGKDD International Conference on Data Mining (KDD'00)*, pages 109–110, August 20–23 2000.

[15] L. Zhang, X. Wu, and Y. Yu. Emergent semantics from folksonomies: A quantitative study. 4090:168–186, 2006.

[1] `http://www.multimedian.nl`