HIERARCHICAL WORD CLUSTERING FOR RELEVANCE JUDGEMENTS IN INFORMATION RETRIEVAL

N. Bassiou, C. Kotropoulos, and I. Pitas

Department of Informatics, Aristotle University of Thessaloniki
Box 451, Thessaloniki 540 06, Greece
{nbassiou, costas, pitas}@zeus.csd.auth.gr
http://poseidon.csd.auth.gr

Abstract. Experiments on document retrieval have been conducted using the baseline information retrieval technique in combination with a hierarchical clustering approach for word clustering. Term reweighting based on document relevance judgements which were automatically determined by the clustering output considerably improved the information retrieval accuracy and performance.

Keywords: document retrieval, word clustering, term reweighting, average recall-precision curves.

1 Introduction

Information retrieval (IR) has been an active research topic the last decades. Nowadays, the term has become synonymous with text retrieval meaning that the task of an IR system is to retrieve documents or texts which are relevant in content to a user's need. Two related but different activities are included in this field. Indexing as a way for the representation of documents, texts or requests, and searching as a way for examining the documents given a query.

Many modules and algorithms were proposed so far for information retrieval from document repositories. In [1, 2, 3] a vector processing model is described in which indexing terms are considered to be coordinates in a multidimensional information space, where documents and queries are represented as vectors. In this space, the ith element that is determined by a term weighting scheme denotes the value of the ith term. The searching part is employed by matching the query against each of the documents in turn. In [4] a probabilistic model is proposed in order to rank the documents of a collection in decreasing order of their probability of relevance to a user's query, a principle known as *probability ranking.* In both techniques, interactive relevance feedback can be used in order to reformulate the original query according to the user judgement on whether the retrieved documents are relevant or irrelevant to his/her information needs. Additional techniques were introduced for automatic-without the user's interference- indexing of documents in topics, most of which are based in document clustering.

In this paper, we build on the information retrieval algorithm proposed in [5] that is based on the probabilistic model. Our effort was to provide a way of automatically determining the relevant and non-relevant judgements for the documents in our collection that are used for term reweighting later. To obtain the relevance judgements we implemented a hierarchical clustering method based on the minimization of the mutual information proposed by Brown in [6]. Although, the clustering algorithm implemented is a greedy algorithm with high computational needs our effort was to demonstrate that term clustering instead of document clustering can also improve the performance of an information retrieval algorithm.

The outline of the paper is as follows. The clustering approach is described in Section 2 and the baseline information retrieval technique together with the description of how clustering results were used in implementing reweighting is briefly presented in Section 3 . Experimental results are illustrated in Section 4 and conclusions are drawn in Section 5.

2 Automatic Word Clustering

Beginning with the common hypothesis in most language models that classes of functionally equivalent words exist, let us assume that given a vocabulary V of size $Q = |V|$, L non-overlapping classes $C_k, k \in [1, \ldots, L]$, exist

$$V = \bigcup_{k=1}^{L} C_k, \qquad C_h \cap C_k = \oslash \text{ for } h \neq k \tag{1}$$

such that the following transition probabilities hold:

$$\forall w^{(i)} \in C_h, \forall w^{(j)} \in C_k, \quad P(w^{(j)}|w^{(i)}) = P(C_k|C_h) \ . \tag{2}$$

The above transition probabilities cannot be exactly estimated since the whole set Φ of all possible sentences is not available. Using a realistic training set $\hat{\Phi}$ such that $|\hat{\Phi}| << |\Phi|$, the estimate of the conditional probability is

$$\hat{P}(w^{(j)}|w^{(i)}) = \frac{N_{\hat{\Phi}}(w^{(i)}, w^{(j)})}{N_{\hat{\Phi}}(w^{(i)})} \tag{3}$$

where $N_{\Phi}(w^{(i)}, w^{(j)})$ is the occurrence of the corresponding bigram in $\hat{\Phi}$.

Since not all the possible bigrams can be seen in a document collection the above probability may be zero. To eliminate the assignment of a zero probability to an unseen bigram, we may use a smoothing technique such as Good Turing, non-linear interpolation or linear interpolation [7]. In non-linear interpolation, (3) is reformulated as follows:

$$\hat{P}(w^{(j)}|w^{(i)}) = \max\{\frac{N_{\hat{\Phi}}(w^{(i)}, w^{(j)}) - D_i}{N_{\hat{\Phi}}(w^{(i)})}, 0\} + D_i \frac{Q - n_0(w^{(i)})}{N_{\hat{\Phi}}(w^{(i)})} P(w^{(j)}) \tag{4}$$

where $n_0(w^{(i)})$ is the number of bigrams that have as predecessor word $w^{(i)}$ and never occurred during training. $P(w^{(j)})$ is the probability of the unigram $w^{(j)}$, and

$$D_i = \frac{Q \cdot b}{n_0(w^{(i)})} \tag{5}$$

where

$$b = \frac{n_1}{n_1 + 2n_2} \tag{6}$$

with n_1 and n_2 being the number of bigrams detected exactly one and two times, respectively in the training set.

We statistically characterize the estimate of transition probability $P(w^{(j)}|w)$ from a given word w to all the other words of the vocabulary, $j = 1, 2, \dots, Q$, in the same way as estimating the probability of the occurrence of Q events, in our case the $(w, w^{(j)})$ bigrams, in $N_{\hat{\Phi}}(w)$ repeated Bernoulli trials [8]

$$\begin{aligned} P(N_{\hat{\Phi}}(w, w^{(1)}), \dots, N_{\hat{\Phi}}(w, w^{(Q)})) = (N_{\hat{\Phi}}(w))! \times \\ \times \prod_{j=1}^{Q} \frac{(N_{\hat{\Phi}}(w, w^{(j)})/N_{\hat{\Phi}}(w))^{N_{\hat{\Phi}}(w, w^{(j)}))}}{(N_{\hat{\Phi}}(w, w^{(j)}))!} \end{aligned} \tag{7}$$

where $P(N_{\hat{\Phi}}(w, w^{(1)}), \dots, N_{\hat{\Phi}}(w, w^{(Q)}))$ denotes the probability of having $N_{\hat{\Phi}}(w, w^{(j)}), j = 1, 2, \dots, Q$ occurrences of the corresponding bigrams in $\hat{\Phi}$.

In (7) since $N_{\hat{\Phi}}(w)$ is sufficiently large and $N_{\hat{\Phi}}(w, w^{(j)})$ is in the $\sqrt{N_{\hat{\Phi}}(w)}$ neighborhood of $N_{\hat{\Phi}}(w, w^{(j)})$, according to De Moivre-Laplace theorem each term in the right-hand side of (7) is approximated by a Q-dimensional Gaussian probability density function (pdf). That is,

$$P(\hat{P}_{\hat{\Phi}}(w, w^{(1)}), \dots, \hat{P}_{\hat{\Phi}}(w, w^{(Q)})) = N(\boldsymbol{\mu}_w, \mathbf{U}_w) \tag{8}$$

where

$$\boldsymbol{\mu}_w = (\hat{P}_{\hat{\Phi}}(w^{(1)}|w), \dots, \hat{P}_{\hat{\Phi}}(w^{(Q)}|w))^T \tag{9}$$

$$\mathbf{U}_w = \frac{1}{N_{\hat{\Phi}}(w^j)} \mathrm{diag}[\hat{P}_{\hat{\Phi}}(w^{(1)}|w), \dots, \hat{P}_{\hat{\Phi}}(w^{(Q)}|w)] \ . \tag{10}$$

In (10) diag[ ] denotes the diagonal matrix having as elements on the main diagonal the indicated arguments. Let us assume that every word $w^{(i)}$ comprises a single class and is represented by an estimated transition probability vector $\mathbf{v}_i$. The probability of the hypothesis H_{pq} that two classes C_p and C_q form a single class is given by

$$\begin{aligned} P(H_{pq}) = \prod_{\forall i \rightarrow w^{(i)} \in C_p \cup C_q} & \frac{1}{(2\pi)^{Q/2} (\det(\mathbf{U}_{pq}))^{1/2}} \\ & \exp\{-\frac{1}{2}(\mathbf{v}_i - \boldsymbol{\mu}_{pq})^T \mathbf{U}_{pq}^{-1} (\mathbf{v}_i - \boldsymbol{\mu}_{pq})\} \end{aligned} \tag{11}$$

where $\boldsymbol{\mu}_{pq}$ and $\mathbf{U}_{pq}$ are, respectively, the mean and covariance of the class formed by merging classes C_p and C_q. In (11), det() denotes the determinant of a matrix and T is the transposition operator. Classes to be merged correspond to the hypothesis that maximizes (11).

By taking the logarithm of (11) and dropping the normalization term we get the following hypothesis

$$H_{pq}^* = \arg\min_{pq}\{\sum_{\forall i \to w^{(i)} \in C_p \cup C_q} (\mathbf{v}_i - \boldsymbol{\mu}_{pq})^T \mathbf{U}_i^{-1} (\mathbf{v}_i - \boldsymbol{\mu}_{pq})\} \tag{12}$$

where $\mathbf{v}_i$ is the vector of the estimated transition probabilities of every word $w^{(i)}$ and its kth component is the transition probability from word $w^{(i)}$ to word $w^{(k)}$, $\hat{P}(w_k|w_i)$. An estimate of μ_{pq} is given by

$$\mu_{pq} = \frac{1}{|C_p| + |C_q|}(\sum_{\forall i \to w^{(i)} \in C_p} \mathbf{v}_i + \sum_{\forall i \to w^{(j)} \in C_q} \mathbf{v}_j) \tag{13}$$

where $|C_p|$ and $|C_q|$ are the numbers of the elements that belong to the corresponding classes.

In (12), we assume that the covariance matrix is diagonal and its kth element is given by

$$[\mathbf{U}_i]_{kk} = \frac{\hat{P}(w^{(k)}|w^{(i)})}{N_{\hat{\Phi}}(w^{(i)})} \quad . \tag{14}$$

Accordingly, (12) is rewritten as

$$H_{pq}^* = \arg\min_{pq}\{\sum_{\forall i \to w^{(i)} \in C_p \cup C_q} \sum_{k=1}^{Q}\{([\mathbf{v}_i]_k - [\boldsymbol{\mu}_{pq}]_k)^2\}[\mathbf{U}_i^{-1}]_{kk}\} \tag{15}$$

and by substituting (14) we obtain

$$H_{pq}^* = \arg\min_{pq}\{\sum_{\forall i \to w^{(i)} \in C_p \cup C_q} \sum_{k=1}^{Q}(\hat{P}(w^{(k)}|w^{(i)}) - \boldsymbol{\mu}_{pq}[k])^2 \cdot \frac{N_{\hat{\Phi}}(w^{(i)})}{\hat{P}(w^{(k)}|w^{(i)})}\} \ . \tag{16}$$

Equation (16) consists the criterion for determining the best pair of classes that should be merged [6].

To summarize, the clustering algorithm developed is described as follows:

- Step 1: Each word of the vocabulary comprises a class on its own. Thus the algorithm starts with Q number of classes.
- Step 2: For each possible class pair the merging criterion defined by (16) is computed.
- Step 3: The two classes that minimize the merging criterion are merged in a single class.
- Step 4: If the number of remaining classes equals a predetermined number of classes L, stop. Otherwise go to Step 2.

In the above-described algorithm, there are approximately $(Q-i)^2/2$ class pairs that have to be examined for merging in each iteration step i. In order to avoid the exhaustive computational needs of this algorithm, we sort the words of the vocabulary in decreasing frequency order and we assign the first $L+1$ words in $L+1$ distinct classes. At each iteration step, we try to find the class pair for which the loss in pointwise mutual information is minimal, we perform the merging acquiring L classes and we insert the next word of the vocabulary in a distinct class resulting again in $L+1$ classes. So at iteration step k, we assign the $(L+k)$th most probable word of the vocabulary in a distinct class and we continue in the same way until no vocabulary words are left. After $Q-L$ steps the words of the vocabulary are assigned in L classes. Using this approach at iteration step i we have to investigate $((L+1)-i)^2/2 < (Q-i)^2/2$ class candidates for merging [6].

If we do not stop the algorithm in L classes but we continue for $Q-1$ merges we obtain a single class containing all the vocabulary words, but the order in which clusters were merged determines a binary tree with the single cluster as root, the words of the vocabulary as leaves, and the intermediate clusters as intermediate nodes.

3 Information Retrieval Technique

Probabilistic information retrieval systems are based on term weighting that are derived from the statistical information about term occurrences in the collection of documents [5]. These weighting schemes deal with either terms or documents, exclusively, or with terms in relation to documents.

Assuming that we have N documents $d_j, 1 \leq j \leq N$, and a total number of terms $t_i, 1 \leq i \leq Q$, where Q is the vocabulary size, the measures described in the following sections are used.

Term or Document Dependent Weighting Measures

The *collection frequency* weight, $CFW(t_i)$, of a term t_i states that terms appearing in few documents are more valuable than those appearing in many, i.e.

$$CFW(t_i) = \log \frac{N}{n} \tag{17}$$

where n is the number of documents containing the term t_i.

The *document length*, $DL(d_j)$, of a document d_j is the total number of different terms in a document.

The *average document length*, $NDL(d_j)$, is the normalized document length and is given by

$$NDL(d_j) = \frac{DL(d_j)}{\overline{DL}(d_j)} \ . \tag{18}$$

Term and Document Dependent Weighting Measures

The *term frequency* weight ,$TF(t_i, d_j)$, of a term t_i in a document d_j is defined as the number of occurrences of this term in the document and implies that a term appearing many times in a document it is likely more important for that document.

The *combined Weight*, $CW(t_i, d_j)$, of a term t_i in a document d_j is the combination of all the above measures. It is described by

$$CW(t_i, d_j) = \frac{CFW(t_i) \times TF(t_i, d_j) \times (k_1 + 1)}{k_1 \times ((1 - b) + (b \times (NDL(d_j))) + TF(t_i, d_j))} \ . \qquad (19)$$

The constant k_1 determines the effect of term frequency in combined weight 19, while the constant b, whose range is the interval $[0, 1]$, controls the effect of document length. Suggested values for these constants are $k_1 = 2$ and $b = 0.75$ [5]. Setting $b = 1$ means that documents are repetitive long, while $b = 0$ means that the documents are long because they are multitopic.

The last measure is the one used for the determination of a document score given a user's request. Given a query the score of each document in which the query terms are present is calculated by adding the combined weights of these terms for the specific document. The documents retrieved to match the user's query are presented ranked according to this score.

A further extension is to exploit additional information about the collected documents that refers to their relevance to a specific topic or request. According to the notion that both the absence and the presence of search terms in a document is equally important, our goal is to accept documents with good terms by rejecting documents without good terms, and to reject documents with bad terms by accepting documents without bad terms. The result then will be the net balance between good and bad terms.

The combination of the presence or absence of a term in a document together with the above notion is given by the components v and u that correspond to presence and absence, respectively. If R is the total number of known relevant documents to a specific topic/request and r is the number of known relevant documents that contain the query term we get for the presence and absence components [9], [10]:

$$v = \log \frac{\frac{r}{R}}{\frac{n-r}{N-R}} \qquad (20)$$

$$u = \log \frac{\frac{R-r}{R}}{\frac{N-n-R+r}{N-R}} \qquad (21)$$

where $n - r$ is the number of irrelevant documents when the search term is present, $N - R$ is the total number of known irrelevant documents, $R - r$ is the number of relevant documents when the search term is absent and $N - n - R + r$ is the number of irrelevant documents where the term is absent. The score for each document is then estimated by adding the presence components for terms present in the document and the absence components for the search terms that are not present in the document.

Reinterpreting the absence/presence components we use the relevance weighting given by

$$w = RW(t_j) = \frac{\frac{r+0.5}{R-r+0.5}}{\frac{n-r+0.5}{N-n-R+r+0.5}} \ . \tag{22}$$

Since according to (22), the good terms have positive weights and the bad terms negative, the effect of scoring for absence is achieved indirectly through the absence of contribution of bad term weights to scores. The equivalence of the two different techniques for relevance weighting, that is, (20)-(21) and (22), is also indicated by the fact that $w = v - u$ holds with 0.5 added for statistical reasons to allow for uncertainty. If we replace the collection frequency weight (17) with the weight given by (22) in combined weight formula defined by (19) we can take into consideration after the first iteration step the relevance of the documents.

To use (22) information with respect to the relevance of documents to queries should be collected. In many systems, the information about whether a document is relevant or irrelevant to a user's request is provided directly by the user and this is known as *relevance feedback.* There are, however, approaches of pseudo-relevance feedback when the documents are annotated beforehand.

By using the clustering approach described in Sect. 2 we can have automatically a rough estimation of R to be used in (22). Considering that each cluster contains terms with similar meaning or words that are most probably found one close to another we can assume that each such cluster comprises a *topic.* Thus the number of documents that are relevant to a specific topic can be found by estimating the number of different documents each term belonging to the cluster appears in. With this assumption we exclude relevant documents that do not contain the search term forcing r to be equal with n. Thus, the contingency table [9] has the form presented in Fig. 3.

Table 1. Indexing and relevance representations

		Relevant	**Documents**	
		+	−	
Documents	+	n	0	n
Indexed	−	$R-n$	$N-R$	$N-n$
		R	$N-R$	N

4 Experimental Results

A number of 288 HTML documents were collected from the web, most of them containing touristic information. These documents were preprocessed in order to remove any HTML tags, symbols or numbers and then they were stemmed using Porter's stemmer in order to replace the words by their stems. The number of resulting stems, and therefore vocabulary size was 8708.

4.1 Clustering Results

Clustering was applied to the stems contained in the collected documents and the 8708 stems of words were partitioned in 669 classes. It is worth mentioning that stems with frequency of appearance in documents less than 3 added too much noise to the clustering procedure causing some classes to have more than 100 elements while normal clusters had an average member of elements around 15 – 20. This noise can easily be ignored since the clustering procedure can be stopped before stems with low frequency are considered for clustering or

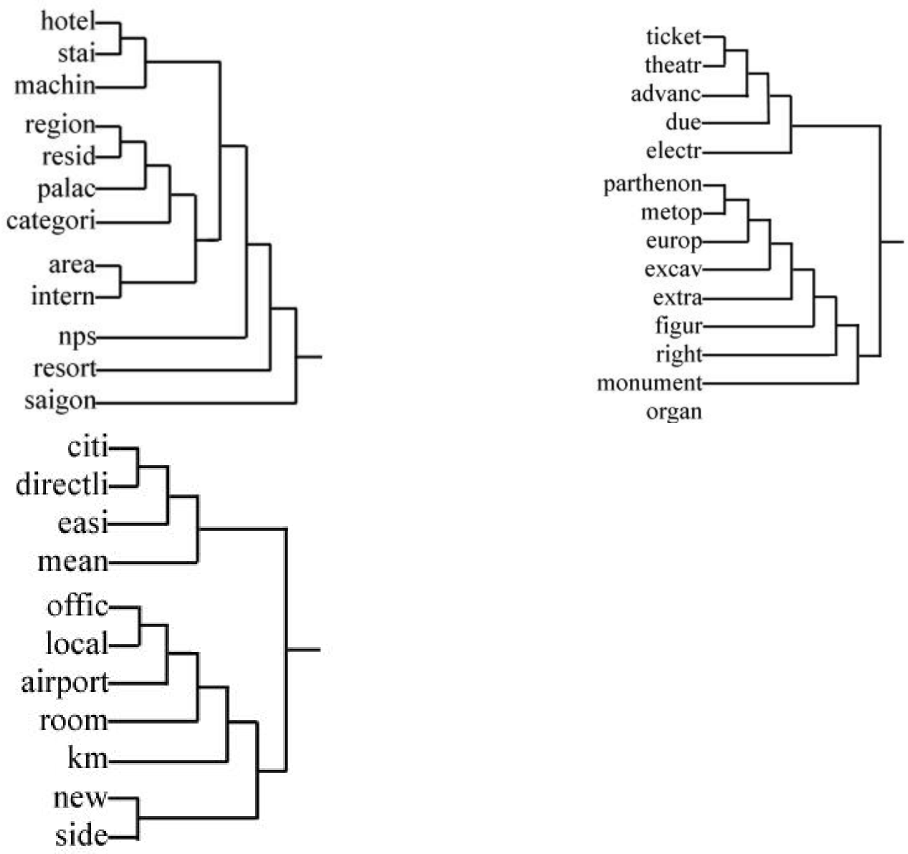

Fig. 1. Sample subtrees of three different clusters

by evaluating the results the algorithm gives before the clustering of the low frequency stems. Some examples of the clustering procedure are presented in Fig. 1 where the screenshots of three clusters are presented.

4.2 Information Retrieval Results

The information retrieval component was trained for the set of the collected stemmed documents and the necessary statistics were estimated according to the equations described in Sect. 3. The implementation of the system involves submitting queries where keywords or phrases are combined by the boolean operators, OR and AND. The OR option forces the documents to be ranked exclusively according to their scores, while the AND option ranks the documents initially according to the number of the query terms they contain and then according to their score. It is observed that the second option is characterized by better recall-precision curves, since a document containing many of the query terms is more likely to be more relevant than a document that contains few query terms but with higher frequencies, that cause the document's score to be quite high.

A set of queries was submitted to the system without taking into consideration the relevance judgements. The same set of queries was submitted but with taking into consideration the relevance weights that were estimated after the clustering procedure. The number of documents in our collection which were relevant to the submitted queries were quite small, around $7 - 10$, since the whole document collection was limited. As a result, we obtained high precision rates, as it can be seen in the average-recall precision curve [11] for the set of the submitted queries as is presented in the Fig. 2. It is worth mentioning that the curve obtained with relevance reweighting lies on top of the curve obtained without relevance reweighting.

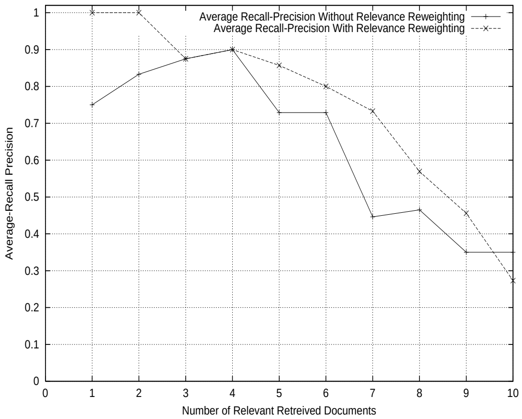

Fig. 2. Average recall-precision curves with and without relevance reweighting

5 Conclusions

In this paper, the use of word clustering in the probabilistic information retrieval technique was presented and its performance has been thoroughly measured with respect to the average-recall precision.

Two algorithms were implemented:

- the minimization of mutual information for word clustering [6], and,
- the probabilistic information retrieval approach with terms reweighting [5].

The results acquired proved that the original information retrieval performance can be improved without the user's interference to relevance judgements. Although the demanding computational needs of the clustering algorithm were known beforehand we used it because, as it was proved, it deals well with the term-based characteristics of the baseline information retrieval approach.

References

[1] Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM **18** (1975) 613–620

[2] Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. In K. Sparck Jones (Ed.), Information Retrieval Experiment (1981)

[3] Salton, G.: The smart environment for retrieval system evaluation-advantages and problem areas. In K. Sparck Jones (Ed.), Information Retrieval Experiment (1981) 316–329

[4] Robertson, S.E.: The probability ranking principle in ir. Journal of Documentation **33** (1977) 126–148

[5] Robertson, S.E., Jones, S.K.: Simple, proven approaches to text retrieval. TR 356 Cambridge University Computer Laboratory May (1977)

[6] Brown, P.F., Della Pietra, V.J., deSouza, P.V., Mercer, R.L., Lai, J.C.: Class based n-gram models of natural language. Computational linguistics **18** no. 2 (1992) 79–85

[7] Beccheti, C., Ricotti, L.P.: Speech Recognition: Theory and C++ Implmentation. J. Wiley, Chichester (1999)

[8] Papoulis, A.: Probability, random variables and stochastic processes, third Edition. McGraw Hill, New-York (1991)

[9] Jones, K.S.: Search term relevance weighting given little relevance information. Journal of Documentation **35** no. I (1979) 30–48

[10] Jones, K.S., Willett, P.: Readings in Information Retrieval. Morgan Kaufman Publishers, San Francisco California (1997)

[11] Manning, C.D., Schutze, H.: Foundations of Statistical Natural Language Processing. MIT Press (1999)