Finding Themes in Medline Documents
Probabilistic Similarity Search

Hagit Shatkay   W. John Wilbur
National Center for Biotechnology Information
National Library of Medicine
National Institutes of Health
Bethesda, Maryland 20984
{*shatkay, wilbur*}@*ncbi.nlm.nih.gov*

Abstract

Large on-line document databases, such as Medline, pose a major challenge of retrieving the few documents most relevant to the user's needs, while minimizing the return rate of non-relevant documents. Retrieval of documents similar to a user-provided example document is a promising query paradigm towards meeting this goal.

We present a new theme-based probabilistic approach for finding documents relevant to a given query document, and summarizing their contents. Preliminary experiments conducted over a subset of Medline documents related to AIDS *demonstrate the effectiveness of our approach.*

1. Introduction

Web-based text databases are rapidly growing, as the *World Wide Web* becomes an increasingly central tool for accessing literature on almost any subject, from recipes to scientific articles. One such database is *Medline*[1]. It is one of the largest, most complete, and most widely used databases for medical documents. It consists of millions of on-line document abstracts, daily updated and queried by thousands of scientists throughout the world. One of the main challenges when maintaining such voluminous databases, is in presenting users with *all* and *only* the documents most relevant to the subject matter they are looking for.

This paper presents a novel approach to searching for a "subject matter" or a *theme* in a large collection of documents, starting from a *single example document*. It is based on the inherent duality in the meaning of the phrase "subject matter": On one hand it is the *set of documents* discussing a certain subject; on the other it is the *set of terms* which are used to describe the subject. As shown in the rest of the paper, we strongly utilize this duality throughout the search process, using an *Expectation Maximization* algorithm to *simultaneously* find the terms representing the *theme* and the documents discussing it. In response to a query we return:

1. An ordered list of *documents* likely to be relevant.
2. An ordered list of *terms* summarizing the theme likely discussed in these documents.

Current query mechanisms over literature databases can be divided into two main categories [17, 23]:

- *Boolean* queries
- *Similarity* queries

When submitting a *boolean* query, the user specifies either a single term (e.g. aids), or a boolean combination of terms (e.g. aids $\wedge$ HIV $\wedge$ Tuberculosis). These terms characterize the subject matter the user is looking for. The result is the set of *all* documents found in the database which satisfy the constraints specified by the query.

This form of query suffers from several deficiencies:

- A *prohibitively large* number of documents are typically retrieved.
- A substantial part of the retrieved documents is *irrelevant* to the query, for a variety of reasons. For instance, irrelevant documents may contain a query term due to its multiple meanings in the language.
- Many relevant documents *may not be retrieved*, despite their relevance, since the terms they contain are semantically *related to* but not *the same* terms as the ones specified in the query. (e.g. *Human Immunodeficiency Virus* as opposed to AIDS).

The last of these limitations, has been addressed in several ways. One approach, stemming from natural language processing, consists of building thesauri of related terms (see for example the work by Pereira *et al.* [13]). An index containing each term in the thesaurus entry, points to all documents containing any of the other related terms. Another successful approach is *latent semantic indexing* [4, 7]. Through the application of singular value decomposition (*SVD*) to a matrix representing a document collection, this method finds semantically related terms in the collection. As in the thesaurus case discussed above, a search for documents containing the term x would result in all the documents containing the terms that are related to x according to the latent semantics analysis.

[1] Medline is maintained by the National Library of Medicine, and can be searched using PubMed, *http://www/ncbi.nlm.nih.gov/PubMed.*

Note that both of these approaches may further aggravate the first two problems inherent to boolean search, since *additional* documents satisfying a query are retrieved, resulting in more potentially irrelevant documents.

By addressing the first of the problems, namely, reducing the set of retrieved documents, to those most relevant to the user's needs, the second problem is also likely to be solved. To achieve such a reduction in the size of the result on one hand, and an increase in its quality and relevance on the other, a shift in the query paradigm is needed.

An alternative paradigm is the use of a *similarity query*, or *query by example*; The user provides a sample document that is relevant, and expects to get back other documents discussing the same subject matter. Various similarity measures over documents have been defined and used in applications of Information Retrieval [8, 15, 17, 22, 23, 24]. We review some of this work in Section 5. However, most existing work does not pay much attention to *explaining* what it is that makes the retrieved documents *similar*. Moreover, in many cases the similarity of the retrieved documents is based on terms that are not necessarily central to the subject matter, resulting in a collection of documents which are similar in some aspects but not the ones sought by the user.

The rest the paper presents our probabilistic approach for finding relevant documents in a database as an instance of the similarity query paradigm. Starting from a single example document, we combine the search for documents bearing the same theme with the search for the terms characterizing this theme. We obtain a set of documents relevant to the subject matter and a set of terms that describe it.

Section 2 introduces the document model and the theme model we use, and lists the assumptions underlying our work. Section 3 provides the algorithm we developed for simultaneously finding both the relevant documents and the relevant terms. Experiments conducted for retrieving topic-specific documents from a database of 32,000 Medline documents discussing HIV, and their results, are described in Section 4. The results demonstrate the effectiveness of our approach for both retrieving and summarizing relevant documents. Section 5 surveys related work. Section 6 outlines on-going work, current applications, and future directions.

2. Models and assumptions

We start with an informal overview of the general framework used, and proceed to present the formal models and assumptions for representing documents, themes and terms.

Our database consists of a large set of documents drawn from a common domain. A domain may be broad (e.g. documents relating to medicine) or be somewhat restricted (e.g. documents discussing pneumonia or HIV). Each document in the database has a *unique numerical identifier*.

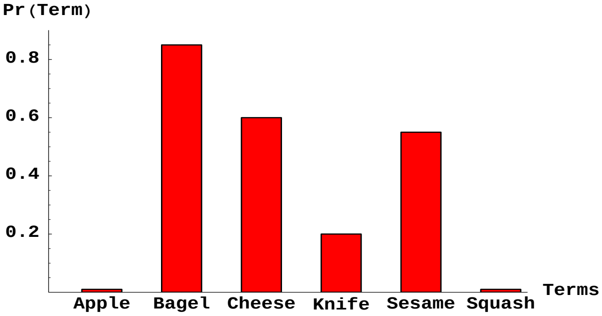

Figure 1: Typical Term Distribution in Bagel Documents

A user looking for documents focused on a certain theme, poses a query by providing the ID number of a document she considers to be representative of the specific theme. As an answer to the query, our goal is to provide the user with:

- a list of documents bearing the same theme as the query document, ordered by their degree of relevance to that theme, and
- a list of terms constituting the theme, ordered by their degree of relevance to the theme.

The idea underlying our probabilistic approach is that a theme can be viewed as a set of independent Bernoulli distributions, one distribution for each term occurring in the database. A document is the result of sampling from such a set of distributions.

To illustrate this idea, consider the documents discussing *bagels* in a large database of documents discussing food. The complete set of terms in the database includes phrases like *apple* and *Squash* which are unlikely to occur in documents discussing Bagels. It also contains terms such as *Cheese*, *Bagel* and *Sesame Seeds* that are highly probable to occur in a bagel document. Thus in the context of "bagel documents" the Bernoulli event of generating the word "Bagel" has, for example, a probability 0.9 while that of generating the word "apple" has a probability of 0.01. Figure 1 demonstrates the distributions of a few terms, plotting terms against their probability to occur in a typical "bagel document"[2].

As a further illustration, consider the complete food database, (denoted as DB), as one large theme – where the theme is "food". In this case, it is easy to find for each term, t_i, a maximum likelihood estimate for its probability (denoted as DB_i) to occur in any "food document", d:

$$\begin{aligned} DB_i &\overset{\text{def}}{=} Pr(t_i \in d | d \in DB) \\ &\approx \frac{\text{\# of documents in DB containing } t_i}{\text{total \# of documents in DB}} \qquad (1) \end{aligned}$$

We shall return to this distribution as part of the formal model presentation.

[2] Note that the probabilities do not sum to 1. Each term corresponds to a separate Bernoulli event.

If one *knows* the characteristic distribution of the Bagel theme, the documents in the database can be ranked according to their likelihood to have been generated by the bagel distribution, and the highest ranking documents are the most likely to be "talking about bagels". However, we do not have such a distribution for ranking documents to begin with. Given many documents discussing bagels, one may be able to obtain sufficient statistics for estimating such a distribution, but all we have is a single example document.

Hence, given a single document, our task is to find this characteristic distribution as well as the documents that are most likely to have been generated by sampling from this distribution. Our algorithm starts by generating a rough approximation of the distribution based on the single example document. It then uses an Expectation-Maximization procedure to iteratively rank the documents based on the current distribution, and generate a new distribution based on the current ranking. The explicit details of the model are given below, and the algorithm itself is discussed in Section 3.

2.1. The document vector model

Let *DB* denote our database of documents.

Let M denote the number of distinct terms $\{t_1, \ldots, t_M\}$ in the whole database. A term, t_i, may be a single word or a longer phrase such as "blood pressure" or "acquired immunodeficiency syndrome". We note that in a standard preprocessing stage, that is not describe here, all stop-words are eliminated and terms consisting of one or two consecutive words are detected and extracted. Thus we are not concerned here with any aspects of term or phrase detection and can assume that the M distinct terms constitute all the terms occurring in the database DB.

A *document*, *d*, in the database is an M-dimensional vector, $\langle d_1,\ d_2,\ \ldots,\ d_M\rangle$, where:

$$d_i = \delta_{di} \overset{\text{def}}{=} \begin{cases} 1 & \text{if } t_i \in d \ , \\ 0 & \text{otherwise} \ . \end{cases} \qquad (2)$$

The document vector is viewed as a result of M independent binomial events. There is assumed some hidden set of M distinct biased coins; each term t_i has associated with it one such biased coin, C_i. When generating a document *d*, for each term t_i we toss the coin C_i. If C_i comes up *Heads* the term t_i is included in *d*, and $d_i = 1$, otherwise d_i is set to 0. In our model, a theme corresponds to a *set* of binomial distributions or "biased coins", as described next.

2.2. The theme model

A *theme*, T, is a *set of documents* discussing a common topic. As demonstrated by the bagel example above, the topic discussed in these documents is modeled by a set of binomial distributions. Each database term, t_i, has a probability p_i^T to occur in documents discussing the topic, and a probability of $(1 - p_i^T)$ to not occur in them.

Formally, p_i^T is a conditional probability defined as:

$$p_i^T \overset{\text{def}}{=} \Pr(t_i \in d | d \in T) \ .$$

Thus, each theme, T, has a separate set of M biased coins, one biased coin for each term in the database. Terms that are highly descriptive of the theme have coins with a high probability of coming up *Heads*, while terms that are unrelated to the theme have a low probability to come up *Heads*.

Given a set of binomial distributions associated with a theme, T, each theme document $d \in T$, is viewed as an instance of sampling from these distributions; In our example, all documents discussing bagels were generated by tossing the set of coins that are biased according to the bagel distribution plotted in Figure 1.

For any given theme, T, there is also a set of complementing distributions governing the documents that are *outside* this theme, $d \notin T$. This is the probability for each term t_i to occur in documents outside the theme set, $DB - T$. It is denoted by q_i^T and defined as:

$$q_i^T \overset{\text{def}}{=} \Pr(t_i \in d | d \notin T) \ .$$

In addition to the above two sets of theme-specific distributions, we also take note of the term distributions in the complete database, DB_i, as defined by Equation 1. This is the probability of each term, t_i, to occur in any document in the database, regardless of its being a theme or an off-theme document, and is easily estimated from the whole database as shown before.

This distribution accounts for the fact that when a document is examined, some terms in it are not meaningful in determining whether it is a topic or an off-topic document; these terms are a result of an arbitrary choice of words by the author. For example, the term "bagel" is highly *likely* to occur in documents discussing bagels; in contrast, the term "apple" is very *unlikely* to occur in documents discussing bagels, although quite likely to occur in documents outside the bagel theme. However, the term "dish" may or may not occur in any document in a food database and in both documents discussing bagels and those not discussing bagel its occurrence is governed by the general database distribution, DB_i, rather than by the distributions p^T or q^T.

Finally, each document d has some prior probability P_d to be in the theme T: $P_d \overset{\text{def}}{=} \Pr(d \in T)$.

Under this model, for a fixed theme T, a database *DB* is viewed as a collection of documents, where each document, *d*, is constructed through the following process of biased sampling, as illustrated in Figure 2:

First a coin, (*OnTheme* in the figure), is tossed to determine if document d is in the theme T. Its probability for coming up $heads$ is P_d.

Then *for each term*, t_i, a term-specific coin, (*FromDB*$_i$ in

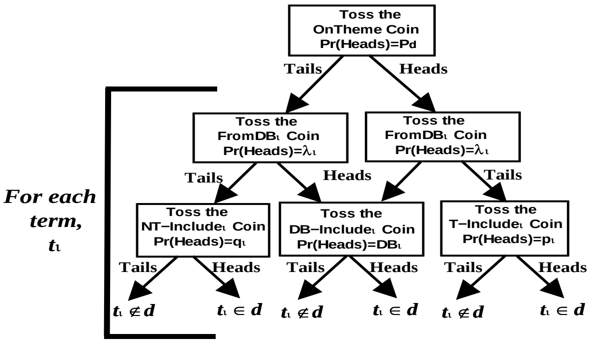

Figure 2: Stochastic Model for Generating Document d.

the figure) is tossed to decide if in document d, t_i is generated according to the general database distribution or according to its specific theme/off-theme distribution. There is one such coin for each term t_i, and its probability of coming up *heads* is λ_i.

Finally, the decision whether to include the term in the document d is based on tossing one of three coins:

- The database coin for term t_i, (*DB-Include*$_i$ in the figure), if t_i is generated according to the *DB* distribution.
- The on-topic coin for term t_i, (*T-Include*$_i$), if d is a theme document, and t_i is generated according to p_i^T.
- The off-topic coin for t_i, (*NT-Include*$_i$), if d is an off-theme document, and t_i is generated according to q_i^T.

Note that we *do know* for each document, $d \in DB$, which terms it contains. Thus, the result of the final events, occurring in the leaves of the chart in Figure 2, are known. However, we *do not* know which document is a *theme* document or which term is generated from the general distribution, DB_i, as opposed to the topic-specific distributions, p_i^T and q_i^T. Hence, the latter events correspond to two sets of *hidden* variables in our model:

- For each document, $d \in DB$, there is a hidden variable Z_d such that

$$Z_d = \begin{cases} 1 & \text{if } d \text{ is a theme document} \quad , \\ 0 & \text{otherwise} \quad . \end{cases}$$

- For each document, d, and term, t_i, there is a hidden variable Z_i^d such that

$$Z_i^d = \begin{cases} 1 & \text{if } d_i\text{'s value is based on } p_i^T \text{ or } q_i^T \quad , \\ 0 & \text{if } d_i\text{'s value is based on } DB_i \quad . \end{cases}$$

To summarize, the complete model, denoted by R, for a specific theme T, consists of the following parameters:

- For each document, d, P_d is the probability of d to be a theme document:

$$P_d \overset{\text{def}}{=} Pr(Z_d = 1) \quad .$$

- For each document d and term t_i, λ_i is the probability that t_i is generated according to the database distribution in document d:

$$\lambda_i \overset{\text{def}}{=} Pr(Z_i^d = 0) \quad .$$

- For each term, t_i, p_i^T is the probability that it occurs in a document d, given that d is a theme document:

$$p_i^T \overset{\text{def}}{=} \Pr(t_i \in d | d \in T) \quad .$$

- For each term, t_i, q_i^T is the probability that it occurs in a document d, given that d is an off-theme document:

$$q_i^T \overset{\text{def}}{=} \Pr(t_i \in d | d \notin T) \quad .$$

- For each term, t_i, DB_i is its probability to occur in a document d within the database, regardless of the theme:

$$DB_i \overset{\text{def}}{=} Pr(t \in d | d \in DB) \quad .$$

2.3. Independence assumptions

To facilitate calculations within the model, we make the following conditional independence assumptions:

- For any two terms, t_i, t_j, their occurrence in a document, d, is *conditionally independent* of each other, *given* the document being a theme/off-theme document. That is: $\Pr(t_i \in d | t_j \in d, Z_d) = \Pr(t_i \in d | Z_d)$.

  This independence assumption allows the probability of generating document d given the value of the variable Z_d, $Pr(d|Z_d)$ to be rewritten as the product:

$$Pr(d|Z_d) = \prod_{i=1}^{M} Pr(\delta_{di}|Z_d)$$

- The variable Z_d and the set $\{Z_i^d\}$ are unconditionally independent (although they do become conditionally dependent given d).
- In the work presented here, P_d is assumed to be independent of the specific document d, and is *the same for all documents d*. That is, a-priori, all documents are equally likely to be in the theme. Obviously, under this assumption we do not fully utilize the one query document known to be in the theme, and the terms occurring in it, for biasing P_d in a useful way. We are currently experimenting with a method that takes advantage of this information. However, the work reported here does not use this method.

Under the above model, the theme finding task reduces to finding the model R that best fits the documents in the database, given that we are looking for a theme, T, based on a query document d_k. (We call d_k the *kernel* document).

At this stage we take note of two interesting properties of our model:

- Documents can be in more than one theme in the database. Given a collection of themes, $T^1, \ldots, T^k$, with respective theme distributions p^{T^j}, q^{T^j}, there is some probability for any document, d, to be in any theme, T^j.
- Unlike most existing work, that deals with complete classification of the database topics [3, 8, 9], we concentrate on finding documents for one particular theme.

The next section describes our algorithm for finding a theme.

3. Probabilistic theme-finding algorithm

Given a kernel document, our task is to find a model R as described above, such that the probability of the model given the database, $\Pr(R|DB)$, is maximized. That is, we want to find the most probable partition of the database into documents that are in the theme and out of the theme, according to the parameters listed in Section 2.2.

Using Bayes rule, the conditional probability above is rewritten as:

$$\Pr(R|DB) = \frac{\Pr(DB|R) \cdot \Pr(R)}{\Pr(DB)} \quad .$$

Since the database is fixed, $\Pr(DB)$ is constant. Also, it is standard to assume that a-priori all models R are equally likely. Hence, the model R maximizing the probability $Pr(R|DB)$ also maximizes the *likelihood* $Pr(DB|R)$, and our task becomes that of finding the maximum likelihood model R.

This is a statistical estimation problem with a lot of missing information (recall the *hidden* variables Z_d and Z_i^d). A general method for addressing such estimation problems is the use of the EM (*Expectation Maximization*) algorithm, developed for hidden Markov models by Baum [2] and generalized by Dempster *et al.* [6]. An EM algorithm starts by initializing the model parameters, R_0, arbitrarily or based on some prior knowledge, and then alternates between:

- the *E-step* of computing the *expected values*, (γ, ξ, and ψ, as defined below), for the hidden variables given the observed database, DB, and the current model R, and
- the *M-step* of finding a new model $\overline{R}$ that maximizes $\Pr(DB|R, \gamma, \xi, \psi)$.

This iterative process is guaranteed, under mild conditions, to provide monotonically increasing convergence of $\Pr(DB|R)$. The algorithm presented here has the same characteristic structure; Starting from a rough estimation of the model parameters, based on the query document, it alternates between the *expectation* step, using the current model to calculate expected values for documents and terms to be in/out of the theme, and the *maximization* step, reestimating model parameters based on the calculated expected values. We have proved our algorithm to be an instance of the EM algorithm, converging to a local maximum of the likelihood function, as part of the derivation of the algorithm. The proof is beyond the scope of this paper, and is not given here. The rest of this section describes our estimation algorithm as an instance of the EM family.

3.1. Initialization

The starting point used in our current implementation is a rough estimate of the parameters, based on the query document. Intuitively, terms t_i that are *rare* in the database but occur in the query document should have a high probability, p_i^T, to occur in the theme, and a low probability λ_i to be generated by the general database distribution. On the other hand, terms occurring in the query document that are also frequent throughout the database should have a high probability, λ_i, to be generated by the database distribution. Terms that are likely to occur in the database but *do not occur* in the query document should have a low p_i^T.

Thus, for each term t_j occurring in the query document, we check the fraction of *DB* documents containing t_j. If fewer than $1/1000$ of the documents in *DB* contain t_j, p_j^T is set to be large, and λ_j is set to be very small (0.002 in the experiments). All terms t_k occurring in more than $1/1000$ of the database documents (regardless of their occurrence in the query document), are assigned λ_k of 0.8, that is – they are likely to have been generated by the database distribution. Frequent terms that do not occur in the query document get a probability p_k^T that is very low (1.0e-100). The probability of all terms to occur outside the theme, q_j^T, is initialized to be the same as their database frequency, DB_j.

The a-priori probability, P_d, for any document d to be a theme document, is fixed at initialization time in the experiments described here, to be 0.001 for all documents in the database.

This initialization strategy is rather coarse and can be further refined, but even this simple scheme leads to good results, as shown in Section 4.

3.2. Estimating model parameters

First, as stated in Section 2, estimating DB_i for each term t_i is straightforward according to Equation 1:

$$DB_i \leftarrow \frac{\text{\# of documents in DB containing } t_i}{\text{\# of documents in DB}} = \frac{\sum_{d \in DB} \delta_{di}}{|DB|} \quad ,$$

where δ_{di} is as defined in Equation 2.

We also recall that the a-priori probability, P_d, is fixed at initialization time to be the same for all documents, and is not reestimated by the algorithm as presented here.

We now describe the estimation of the other parameters, namely p_i^T, q_i^T and λ_i, starting with the *Maximization* step and following by some of the detail of the *Expectation* step.

Let R be the current model, consisting of the sets of parameters $\{p_i^T\}$, $\{q_i^T\}$ and $\{\lambda_i\}$, and $\overline{R}$ be the reestimated sets of parameters $\{\overline{p}_i^T\}$, $\{\overline{q}_i^T\}$ and $\{\overline{\lambda}_i\}$.

Suppose we have executed the *Expectation* step, deriving the following expected values, for each term, t_i and document, d:

γ_{di} – The *expected* value of the random variable $(1 - Z_i^d)$, which is the expected value of the event that term t_i is generated according to the *general database distribution*, DB_i, in document d. Formally:

$$\gamma_{di} \overset{\text{def}}{=} \Pr(Z_i^d = 0 | d, R) \ ,$$

ξ_{di} – The expected value of the joint event of document d being a *theme document*, and term t_i generated in it according to the *theme-specific distribution* in document d. Formally:

$$\xi_{di} \overset{\text{def}}{=} \Pr(Z_d = 1 \wedge Z_i^d = 1 | d, R) \ .$$

ψ_{di} – The expected value of the joint event of document d being an *off-theme document*, and term t_i generated in it according to the *off-theme-specific distribution* in document d. Formally:

$$\psi_{di} \overset{\text{def}}{=} \Pr(Z_d = 0 \wedge Z_i^d = 1 | d, R) \ .$$

Once the above values are calculated, the model parameters are reestimated as:

$$\overline{\lambda}_i \leftarrow \frac{\text{Expected\#of database documents in which } t_i \\ \text{is generated according to the database distribution}}{\text{Total \# of documents in the database}} \ ,$$

$$\overline{p}_i^T \leftarrow \frac{\text{Expected \# of theme documents } \textit{containing } t_i\text{, in which } t_i \\ \text{is generated according to the theme/off-theme distribution}}{\text{Expected \# of theme documents in which } t_i \\ \text{is generated according to the theme/off-theme distribution}} \ ,$$

$$\overline{q}_i^T \leftarrow \frac{\text{Expected \# of off-theme documents } \textit{containing } t_i\text{, in which} \\ t_i \text{ is generated according to the theme/off-theme distribution}}{\text{Expected \# of off-theme documents in which } t_i \\ \text{is generated according to the theme/off-theme distribution}} \ .$$

The explicit update formulae are therefore:

$$\overline{\lambda}_i \leftarrow \frac{\sum\limits_{d \in DB} \gamma_{di}}{|DB|} \ , \ \overline{p}_i^T \leftarrow \frac{\sum\limits_{d \in DB} \xi_{di} \cdot \delta_{di}}{\sum\limits_{d \in DB} \xi_{di}} \ , \ \overline{q}_i^T \leftarrow \frac{\sum\limits_{d \in DB} \psi_{di} \cdot \delta_{di}}{\sum\limits_{d \in DB} \psi_{di}} \ .$$

Again, δ_{di} is the indicator function defined in Equation 2.

3.3. Calculating the expected counts

It is now left to calculate the expected values γ_{di}, ξ_{di}, and ψ_{di}. The derivation is done through a straightforward application of Bayes rule and standard algebraic manipulations, and for the sake of brevity we omit most of it here. As an illustrative example, the derivation of $\gamma_{di} = \Pr(Z_i^d = 0 | R, d)$ is given.

Using the assumption stated earlier that terms within a document d are *independent* of each other, we can rewrite:

$$\gamma_{di} \overset{\text{def}}{=} Pr(Z_i^d = 0 | R, d) = Pr(Z_i^d = 0 | R, \delta_{di}) \ .$$

By Bayes rule:

$$Pr(Z_i^d = 0 | R, \delta_{di}) = \frac{Pr(\delta_{di} | Z_i^d = 0, R) \cdot Pr(Z_i^d = 0 | R)}{Pr(\delta_{di} | R)} \ .$$

By definition of DB_i and λ_i, and by decomposing $Pr(\delta_{di} | R)$ according to the explicit values of Z_i^d, the right-hand-side of the above equation is rewritten as:

$$Pr(Z_i^d = 0 | R, \delta_{di}) = \frac{{DB_i}^{\delta_{di}} (1 - DB_i)^{1 - \delta_{di}} \cdot \lambda_i}{\sum\limits_{l \in \{0,1\}} Pr(\delta_{di} | Z_i^d = l, R) \cdot Pr(Z_i^d = l | R)} \ . \tag{3}$$

The first summand in the denominator of (3) is the same as the numerator, while the second is decomposed as:

$$\begin{aligned} & Pr(\delta_{di} | Z_i^d = 1, R) \cdot Pr(Z_i^d = 1 | R) \\ = & \sum_{l \in \{0,1\}} Pr(\delta_{di}, Z_d = l | Z_i^d = 1, R) \cdot (1 - \lambda_i) \\ = & \sum_{l \in \{0,1\}} \big[Pr(\delta_{di} | Z_d = l, Z_i^d = 1, R) \cdot \\ & \qquad Pr(Z_d = l | Z_i^d = 1, R) \big] \cdot (1 - \lambda_i) \qquad (4) \\ = & \ \big[(p_i^T)^{\delta_{di}} \cdot (1 - p_i^T)^{(1 - \delta_{di})} \cdot p_d \ + \\ & (q_i^T)^{\delta_{di}} \cdot (1 - q_i^T)^{(1 - \delta_{di})} \cdot (1 - p_d) \big] \cdot (1 - \lambda_i) \ , \qquad (5) \end{aligned}$$

where the rewrite of expression 4 as 5 uses the independence of Z_i^d and Z_d when d is *not* given. Similar derivation is used for estimating ξ_{di} and ψ_{di}. The whole process of calculating the expected values, and reestimating the model parameters is iterated until the parameters do not (significantly) change and convergence is reached.

4. Experiments and results

To test the algorithm, we applied it to a subset of Medline, consisting of 32,000 abstracts discussing AIDS. Standard stop words and terms that are very frequent (appear in more than $1/10$ of the documents) were omitted from the text. We then picked 10 document abstracts out of this set, each discussing some complication associated with AIDS. The documents were picked from a list returned from a boolean search for specific complications, rather than completely at random. This is a reasonable testbed, since a typical user, looking for information based on an example article is likely to provide a "content-bearing" example and not merely a "random" one. Note that the 10 documents were picked based on their titles alone, without examining their contents.

Each of the 10 documents was used in turn as a query kernel, and our algorithm was used to find a *theme* based on it. We find the set of relevant documents, as well as the relevant terms, starting from this one document. From now on,

Failure of screening to detect HIV in a foreign laborer who died of Toxoplasmosis of the central nervous system.
The most common neurological complication in patients with acquired immunodeficiency syndrome (AIDS) is cerebral toxoplasmosis. Patients with cerebral toxoplasmosis have characteristic findings on clinical examination and neuroimaging. They require prolonged treatment and have a considerable mortality rate. We report a case of cerebral toxoplasmosis in a foreign laborer with AIDS, in whom a human immunodeficiency virus (HIV) screening test failed to detect-HIV infection. The patient, a 23-year-old man from Thailand, presented in a confused state 2 weeks after his arrival in Taiwan. Computed tomography showed a mass effect, and magnetic resonance imaging showed multiple ring-enhanced lesions in the cerebrum. Serologic tests were positive for anti-HIV antibody and also showed high anti-Toxoplasma immunoglobulin G titers. Although symptomatic treatment was initiated, the patient's condition deteriorated rapidly and he died of multiple organ failure due to brain stem herniation a few days after admission. As the number of foreign laborers working in Taiwan has increased dramatically in recent years, the issues raised by this case are the efficacy of our screening protocols for foreign laborers and the increased occupational hazards encountered by medical personnel in Taiwan.

Expression and antigenicity of human herpesvirus 8 encoded ORF59 protein in AIDS-associated Kaposi's sarcoma.
Human herpesvirus 8 (HHV-8, Kaposi's sarcoma-associated herpesvirus, KSHV) is a new herpes virus isolated from patients with AIDS-associated Kaposi's sarcoma (AIDS-KS). The ORF59 protein of HHV-8 has recently been shown to encode a processivity factor (PF-8) for HHV-8-encoded DNA polymerase. By immunoscreening a cDNA library derived from the HHV-8-infected cell line TY-1, ORF59 antigen was identified in AIDS-KS patients. Immunoblotting revealed that recombinant ORF59 protein reacted with sera from patients with AIDS-KS. Enzyme-linked immunosorbent assay (ELISA) using ORF59-recombinant protein as the antigen revealed that 7 of 22 (31.8%) AIDS-KS patients and 6 of 263 (2.2%) Japanese HIV-negative patients or healthy blood donors were positive for anti-ORF59 antibodies. Immunohistochemistry using anti-ORF59 rabbit antibodies revealed that this protein was expressed in some of the tumor cells found in KS tissues and that ORF59 protein was detected in 11 of 22 (50%) AIDS-KS tissues. In situ hybridization indicated that some of KS tumor cells were positive for HHV-8 T1.1 mRNA in the same specimen. These data suggest that ORF59 is one of the HHV-8 encoded antigens in patients with AIDS-KS and also indicated that viral replication occurred in some of KS tumor cells.

Figure 3: Two of the abstracts used as kernels for our algorithm

we refer to each of the 10 query documents, around each of which a theme was generated, as a *kernel* document.

Figure 3 shows the titles and abstracts for 2 of the 10 kernel documents. The document on the left discusses screening failure in an extreme case of Toxoplasmosis, which is a severe infection associated with AIDS, effecting the central nervous system. It is often detected by the presence of a typical ring pattern in the brain image. The document on the right discusses Kaposi's Sarcoma, a skin cancer common in AIDS patients. In particular, the paper discusses genetic aspects of Kaposi's Sarcoma, related to Herpesvirus-8.

For each of the 10 themes generated from the 10 kernels by our algorithm, we list the top ranking documents, where the ranking is based on the documents probability to be in the theme, $Pr(Z_d = 1|d, DB, R)$. The higher this probability – the more likely a document is to be a theme document.

Figure 4 shows the titles for the 4 highest ranking documents[3] for each of the two kernels of Figure 3. The top document corresponds in both examples to the query document itself, but this is not necessarily always the case. At times, the query document may not be the strongest representative of its own theme, causing other documents, that are highly relevant to the same theme, to rank higher.

Taking a closer look at the results for the Toxoplasmosis-related documents on the left of Figure 4 highlights some of the strengths of our algorithm:

Note that the title of the second document is concerned with diagnosis problems of complications in the central nervous system, other than Toxoplasmosis. The document itself discusses a typical ring visible in brain MRI images, which is an indicator for brain tumor as well as for other aids-related infections. This same ring is also discussed in the kernel document, since it is also an indicator for Toxoplasmosis (see Figure 3 left). Our algorithm links the kernel document to that other document, despite the fact that the latter does not discuss Toxoplasmosis. This link can alert physicians about a possible mistake in their diagnosis, if the diagnosis is based on the ring observed in the brain image.

The third document in the theme discusses encephalitis which is an inflammation of the brain, detected in aids patients due to infection other than Toxoplasmosis (Trypanosoma cruzi). Again, the paper warns about misdiagnosis, due to similar brain image pattern in Toxoplasmosis as in Trypanosoma cruzi.

In the Kaposi's Sarcoma case, the kernel document specifically discusses herpesvirus 8, as it relates to Kaposi's Sarcoma. The documents ranking 3^{rd} and 4^{th} (as well as several other high ranking documents not shown here) indeed discuss this specific topic. On the downside we note that the document ranked second, discusses rare cases of bone-based Kaposi's Sarcoma. The main reason for its high ranking despite its relative irrelevance, is it's being very short, consisting mostly of generic Kaposi's Sarcoma-related terms, and little else. There are relatively few docu-

[3] Limiting the presentation to the 4 top documents is for illustration purposes only. Typically, documents ranking lower than that are still highly relevant to the kernel document.

Failure of screening to detect HIV in a foreign laborer who died of toxoplasmosis of the central nervous system.

AIDS-associated cytomegalovirus infection mimicking central nervous system tumors: a diagnostic challenge.

Chagasic granulomatous encephalitis in immunosuppressed patients. Computed tomography and magnetic resonance imaging findings.

Isolated homonymous lateral hemianopsia revealing central nervous system toxoplasmosis as the initial manifestation of AIDS.

Expression and antigenicity of human herpesvirus 8 encoded ORF59 protein in AIDS-associated Kaposi's sarcoma.

Primary intraosseous AIDS-associated Kaposi's sarcoma. Report of two cases with initial jaw involvement.

Expression of human herpesvirus-8 (HHV-8) encoded pathogenic genes in Kaposi's sarcoma (KS) primary lesions

Further confirmation of the association of human herpesvirus 8 with Kaposi's sarcoma.

Figure 4: Titles of the 4 top documents retrieved for both kernels

ments in our database discussing herpesvirus together with Kaposi's sarcoma, compared with the large number of documents discussing other aspects of Kaposi's sarcoma. Documents whose few terms are dominant Kaposi's sarcoma terms, gravitate toward any specific theme related to Kaposi's sarcoma, and bias it towards generalization. Controlling the search to avoid such generalization is among the issues we are currently investigating.

The other part of the output is a list of *terms* representative of the theme. Note that simply picking the terms with highest probability p_i^T is not a good strategy. The terms most frequent in the theme are probably the same as those most commonly occurring in the whole database, thus are not good representatives of the theme. To overcome this, we generate a list of terms that are most likely to occur in theme documents *and* be generated by the theme distribution ($\Pr(Z_d = 1, Z_i^d = 1|d,t) \geq 0.6$), as well as terms t_i for which p_i^T is much larger than DB_i, that is, terms that are much more probable in the theme than outside the theme. Out of this restricted list we pick the ones with the highest probability p_i^T, to occur in the theme. We print these terms ordered by the ratio p_i^T/q_i^T so that the top terms are the ones most distinguishing the topic documents from the off-topic documents. An example of the lists for the two themes learned starting from the kernel documents of Figure 3 is shown in Table 1.

It is easily seen that the terms indicating the general contents of each theme, appear on the list (*toxoplasmosis, kaposi's sarcoma, kshv* – a shorthand for *Kaposi's Sarcoma Herpes Virus*). Moreover, theme-specific words such as *magnetic resonance, nervous system* etc. in the Toxoplasmosis case, and *human herpesvirus/hhv* in the Kaposi's Sarcoma case, are dominant as well. (Recall that the Toxoplasmosis documents deal with brain image analysis.)

Toxoplasmosis theme	Kaposi's Sarcoma theme
toxoplasmosis	associated herpesvirus
resonance imaging	kshv
nervous system	sarcoma associated
nervous	human herpesvirus
central nervous	kaposi's sarcoma
cerebral toxoplasmosis	kaposi's
magnetic resonance	herpesvirus
old man	sarcoma
central	hhv
year old	aids associated

Table 1: Top 10 terms for each theme, ordered by decreasing ratio p_i^T/q_i^T

Table 1 contains some terms that are not informative (such as "year old" and "old man" – both subcomponents of the phrase "*x year old man*") due to their high likelihood to occur in case-report documents. Also, some grammatically-correct terms, (e.g. "related herpesvirus"), are semantically useless, and their occurrence in the term list is redundant. We note, however, that a human expert looking at such a report can easily distinguish the content-bearing terms from the others. Future work will concentrate on ways to make the term summary more descriptive of the specific topic, eliminating non-informative terms, and on algorithms for extracting meaningful phrases from the text. The use of such methods is expected to further improve the presentation of themes to the user.

The reported results demonstrate the ability of our algorithm to construct a set of documents with a common theme, along with a content summary, starting from a single example document. Experiments performed on other document sets both in and out of Medline produce similar results.

5. Related work

The work presented here is concerned with finding themes in a database of documents, based on a single example. Our method essentially results in soft clustering of the database into *two sets*, namely, the documents that bear the theme and those that do not. We provide a brief survey of the work on document clustering which bears some resemblance to ours, and also review some work related to finding key words and hidden semantics in documents, mostly pertaining to boolean queries.

Clustering techniques can be divided into two main categories: *supervised* and *unsupervised*. In the supervised case, usually referred to as *classification*, a training set of documents, labeled by their respective classes, is provided. From this data, rules for classifying unseen documents are learned. These rules are then applied to yet unclassified documents in order to form complete classes of documents based on some predefined labels. See for instance work by Koller and Sahami [10, 11] for discussion of this approach.

In the unsupervised case (which is closer to our work) the complete set of documents is partitioned into sets of interrelated clusters based on various metrics over documents and over sets of documents. The underlying idea is to keep similar documents within the same clusters, and have the clusters themselves as distinctive as possible from each other. Despite Voorhees' [21] claim that little is gained from using clusters for improving retrieval when strict partition is enforced over the documents set, a lot of work on document clustering was performed during the last decade. Some of it concentrating on "soft", probabilistic clustering, in which documents might be assigned to more than one cluster, with a probability distribution governing the assignment. Various clustering algorithms such as K-means [3], hierarchical agglomeration [8], statistical and multi-valued mixture models [18, 16] have been applied to documents, mostly in an attempt to build a complete hierarchy of documents.

Another work based on clustering similar documents without building a complete hierarchy is an earlier work, done by the second author, on *neighboring* in the context of Medline [22]. In this case documents that are close together based on a probabilistic variant of the *cosine coefficient* are clustered into a single neighborhood, and the neighborhood of document d is retrieved whenever the user looks for documents similar to d. Our experience with this approach showed that in many cases, the neighboring algorithm pulls together documents based on irrelevant terms, and due to the nature of the algorithm a document viewed as a neighbor never leaves the neighborhood. In contrast, our iterative algorithm lets probabilities adapt, allowing documents to dynamically become more or less likely to be in the theme. Thus documents that seem related to the kernel document in early iterations may not rank high in later iterations (and vice-versa), depending on which other documents are considered as highly relevant.

The work most closely related to ours in the clustering domain is the recent work on the *cluster-abstraction model* by Hofmann [9]. He uses EM to build a hierarchy of topic-based classes, while finding the meaningful words in clusters. His document model is based on the *multinomial* distribution rather than the binomial. The *multinomial* model is often used for representing full-text documents in which words occur multiple times. In a database of abstracts, like Medline, a multinomial model is not as appropriate. Moreover, it prevents multiple terms from having high probability since the mass over all terms has to sum to 1. In addition, the task Hofmann addresses is that of generating a *complete hierarchy* of clusters rather than finding documents related to a particular topic. The main drawback for a complete precalculated clustering is that in a very large database each cluster is typically prohibitively large for a user to browse through; the smaller clusters correspond to very specific topics – requiring highly sophisticated queries to be specified by the user, (or an interactive dialog), in order to find them. So far, there has been no successful attempt applying unsupervised clustering to realistically large document sets.

Clustering into two sets – the theme and off-theme documents – is intuitively a more manageable task, and therefore can be expected to be more easily achieved even for large data sets. Our approach allows documents to be strongly associated with multiple themes, multiple terms to be associated with a theme and multiple themes to be associated with a term. (Note that even under soft complete clustering, a document associated with multiple classes can only be "a little bit" associated with each of the classes, since the probabilities must sum to 1).

Other work in the information retrieval community relates to ours in the context of finding terms denoting common topics in related documents. Work on Latent Semantics Indexing (LSI) [5] deals with finding terms related to a document even when not explicitly occurring in it. This method is used when a collection of related documents is given, and is useful for improving boolean queries, by finding documents that are not explicitly mentioning the query terms but are still relevant to it. So far this method has not been applied to large collections of documents.

Work by Croft *et al.* [20, 14] concentrated on the use of Bayesian networks for representing documents and indexing them based on terms likely to be important in them, but their approach requires a lot of unavailable information to be obtained in order to rigorously construct such networks.

Another related issue is that of automatic summarization and finding content-bearing words in text documents. Recent work by Barzilay *et al.* [1] concentrates on summarizing related documents by finding common phrases in them, in the context of news reports. This method can not be readily applied to Medline abstracts in which identical meaningful phrases are typically rare. Work by Marx *et al.* [12] takes an initial step towards finding terms which make a set of similar documents "similar". Their work is based on the existence of a given metric for measuring the similarity between terms in documents. Both of these methods assume that a set of related articles already exists, awaiting summarization. It is important to note that our approach *does not* separate the task of finding themes into the two stages of first finding the documents and then extracting the words summarizing them, but rather we simultaneously build the set of documents and the set of characteristic terms.

6. Conclusion and future work

This paper presented a new theme-generation approach for obtaining relevant documents along with a summary justifying their relevance, based on an example document. We have applied the algorithm to document collections other than the one presented here, (e.g. a standard collection of Reuters articles), with similar success. Currently, we are experimenting with a variety of initialization methods, and addressing the issue of quantitative assessment of the results relative to a human expert.

In a particularly promising new application, [19], we use the retrieval and summarization algorithm described here to help in the analysis of gene expression arrays, through automated mining of the relevant bio-medical literature.

Acknowledgments

We thank Carolina Nadel, M.D, for her help in the interpretation of the AIDS-related results, and Luis Ortiz for his advice on EM issues.

References

[1] R. Barzilay, K. R. McKeown and M. Elhadad, *Information Fusion in the Context of Multi-Document Summarization*, In Proc. of the 37^{th} Meeting of the Assoc. for Computational Linguistics, 1999, 550–557.

[2] L. E. Baum, T. Petrie, G. Soules and N. Weiss, *A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains*, The Annals of Mathematical Statistics, 41 , no. 1, 1970, 164–171.

[3] 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 Proc. of the SIGIR Conf. on Research and Development in Information Retrieval, 1995.

[4] S. Deerwester *et al.*, *Indexing by Latent Semantic Analysis*, Journal of the Society for Information Science, 41 , no. 6, 1990, 391–407.

[5] S. T. Dumais *et al.*, *Using Latent Semantic Analysis to Improve Access to Textual Information*, In Proc. of the Conf. on Human Factors in Computing (CHI88), 1988.

[6] A. P. Dempster, N. M. Laird and D. B. Rubin, *Maximum Likelihood from Incomplete Data via the EM Algorithm*, J. of the Royal Statistical Society, 39 , no. 1, 1977, 1–38.

[7] S. T. Dumais, *Enhancing Performance in Latent Semantic (LSI) Indexing*, Behavior Research Methods, Instruments and Computers, 23 , no. 2, 1990, 229–236.

[8] M. Goldszmidt and M. Sahami, *A Probabilistic Approach to Full-Text Document Clustering*, Tech. Report ITAD-433-MS-98-044, SRI International, 1998.

[9] T. Hofmann, *The Cluster-Abstraction Model: Unsupervised Learning of Topic Hierarchies from Text Data*, In Proc. of the Int. Joint Conf. on Artificial Intelligence, 1999.

[10] D. Koller and M. Sahami, *Toward Optimal Feature Selection*, In Proc. of the Int. Conf. on Machine Learning, 1996.

[11] D. Koller and M. Sahami, *Hierarchically Classifying Documents Using Very Few Words*, In Proc. of the Int. Conf. on Machine Learning, 1997.

[12] Z. Marx, I. Dagan and E. Shamir, *Detecting Sub-Topic Correspondence through Bipartite Term Clustering*, In Proc. of the Workshop on Unsupervised Learning in Natural Language Processing, 1999, pp. 45–51.

[13] F. Pereira, N. Tishby and L. Lee, *Distributional Clustering Of English Words*, In Proc. of the Meeting of the Assoc. for Computational Linguistics, 1993, 183–190.

[14] T. B. Rajashekar and W. B. Croft, *Combining Automatic and Manual Index Representations in Probabilistic Retrieval*, J. of the American Society for Information Science, 46 , no. 4, 1995.

[15] S. E. Robertson and S. Walker, *Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval*, In Proc. of the SIGIR Conf. on Research and Development in Information Retrieval, 1994.

[16] M. Sahami, *Using Machine Learning to Improve Information Access*, PhD thesis, Stanford University,Computer Science Department, 1998.

[17] G. Salton, *Automatic Text Processing*. Addison-Wesley, 1989.

[18] M. Sahami, M. Hearst and E. Saund, *Applying the Multiple Cause Mixture Model to Text Categorization*, In Proc. of the Int. Conf. on Machine Learning, 1996.

[19] H. Shatkay, S. Edwards, W. J. Wilbur and M. Boguski, *Genes, Themes and Microarrays*, Submitted to the Int. Conf. on Intelligent Systems in Molecular Biology, 2000.

[20] H. Turtle and W. B. Croft, *Inference Networks for Document Retrieval*, In Proc. of the SIGIR Conf. on Research and Development in Information Retrieval, 1990, pp. 1–24.

[21] E. M. Voorhees, *The Cluster Hypothesis Revisited*, In Proc. of the SIGIR Conf. on Research and Development in Information Retrieval, 1985, pp. 188–196.

[22] W. J. Wilbur and L. Coffee, *The Effectiveness of Document Neighboring in Search Enhancement*, Information Processing and Management, 30 , no. 2, 1994, 253–266.

[23] I. H. Witten, A. Moffat and T. C. Bell, *Managing Gigabytes, Compressing and Indexing Documents and Images*. Morgan-Kaufmann, 2 edition, 1999.

[24] Y. Yang and X. Liu, *A re-examination of text categorization methods*, In Proc. of the SIGIR Conf. on Research and Development in Information Retrieval, 1999.