

Topic-Specific Link Analysis using Independent Components for Information Retrieval

Wray Buntine, Jaakko Löffström, Sami Perttu and Kimmo Valtonen

Helsinki Institute for Information Technology (HIIT)

P.O. Box 9800, FIN-02015 HUT, Finland

First.Last@Hiit.FI

Abstract

There has been mixed success in applying semantic component analysis (LSA, PLSA, discrete PCA, etc.) to information retrieval. Previous experiments have shown that high-fidelity language models do not imply good quality retrieval. Here we combine link analysis with discrete PCA (a semantic component method) to develop an auxiliary score for information retrieval that is used in post-filtering documents retrieved via regular Tf.Idf methods. For this, we use a topic-specific version of link analysis based on topics developed automatically via discrete PCA methods. To evaluate the resultant topic and link based scoring, a demonstration has been built using the Wikipedia, the public domain encyclopedia on the web.

Introduction

More sophisticated language models are starting to be used in information retrieval (Ponte & Croft 1998; Nallapati 2004) and some real successes are being achieved with their use (Craswell & Hawking 2003). A document modelling approach based on discrete versions of PCA (Hofmann 1999; Blei, Ng, & Jordan 2003; Buntine & Jakulin 2004) has been applied to the language modelling task in information retrieval (Buntine & Jakulin 2004; Canny 2004). However, it has been shown experimentally that this is not necessarily the right approach to use (Azzopardi, Girolami, & van Risjbergen 2003). The problem can be explained as follows: when answering a query about “german immigration,” a general statistical model built on a full international news corpus often lacks the fidelity on these two key words combined. Ideally, one would like a statistical model more specifically about “german immigration,” if it were feasible. Thus the statistically based language modelling approach to information retrieval is still needing of development.

Here we take an alternative path for using statistical models, such as those built with discrete PCA, into information retrieval. Our approach is motivated by the widespread observation that people would like to be able to bias their searches towards specific areas, but they find it difficult to do in general. Web critics have reported that Google, for instance,

suffers perceived bias in some searches because of the overriding statistics of word usage in its corpus (“the web”) in contrast with their dictionary word senses (Johnson 2003): on the internet an “apple” is a computer, not something you eat, and “Madonna” is an often-times risqué pop icon, not a religious icon. Thus one wants to use a keyword “Madonna” but bias the topic somehow towards Christianity in order to get the religious word sense.

A major user interface problem here is that people have trouble navigating concept hierarchies or ontologies (Suomela & Kekäläinen 2005), especially when they are unfamiliar with them. This user interface problem is con-founded because good topic hierarchies and ontologies are usually multi-faceted and search might require specifying multiple nodes. Thus we combine the following techniques:

Topic by example: Users do not have to know the hierarchy, or browse it, or travel to multiple places to get multi-facets for their search. They just enter a few words describing their general topic area in a “context words” box and let the system work out the topics “by example”. An example of the input screen is shown in Figure 1.

Topic specific page-rank: Many pages can be topically relevant, but when dealing with a specific topic area or combination of topic areas, which pages are considered the most important in terms of topically relevant citations? Topic specific versions of page rank (Haveliwala 2002; Richardson & Domingos 2002) address this.

Result filtering: The top results from a regular Tf.Idf query are reranked using a weighted combination of topic-specific page rank. In this way, the choice of topic “by example” affects the results.

Here we first apply the discrete PCA method to develop topics automatically. This gives topics suitable for the corpus, and a multi-faceted classification of all pages in it. We then apply these using a topic-specific version of page rank (Richardson & Domingos 2002) that is based on the notion of a random surfer willing to hit the back button when a non-topical page is encountered. This gives topic specific rankings for pages that can be used in a topic-augmented search interface.

Our intent is that these techniques yield a secondary topical score for retrieval in conjunction with a primary keyword based score such as Tf.Idf. Thus relevance of a docu-



Figure 1: The search options on the results page

ment is a combination of both keyword relevance and topical relevance. Because search users are usually daunted by anything more than just a keyword box, and because keyword search currently works quite well, our default is to make the keyword entry and the topical entry equivalent initially in a search, and only give the option to change the topic, as shown in Figure 1, after a first batch of results have been returned. Thus the initial search screen contains no “context words” box.

Our platform for experiments with these methods is the English language part of the Wikipedia¹, an open source Encyclopedia. This has a good internal link structure necessary for the link analysis. The system is demonstrated² at our test website (<http://kearsage.hiit.fi/wikisearch.html>). The website is being used to test interface concepts as well as perform user studies.

The combination of topic-specific and link-based scoring is fundamental, we believe, to the success of this method. Topic-based scoring alone can return documents with high topical scores, but they are not “characteristic” documents for the topic and keyword combination, rather they are “typical”. A document with high topical content is not necessarily characteristic. For instance, entering the query “madonna” gives the following pages titles as top results under a standard OKAPI BM25 version of Tf.Idf, under Google, and under our system (“Topical filtering”). These are listed in rank order:

Tf.Idf: Madonna (entertainer), Unreleased Madonna songs, List of music videos by year work in progress, Bedtime Stories (Madonna), American Life

Google: Madonna (entertainer), Madonna (singer), Madonna, Unreleased Madonna Songs, Black Madonna

Topical filtering: Madonna, Madonna (entertainer), Unreleased Madonna songs, The Madonna, American Life

Tf.Idf essentially returns documents with many instances of the word Madonna. Google essentially returns documents voted by web-links as being most important, mostly Madonna the entertainer. Our approach sees Madonna as a word with both entertainment and religious connotations, and returns important documents with a better topical mix. “Madonna” in this case is the main disambiguating page that points to the different versions of Madonna. It becomes the highest ranked using our topical filtering due to it being a

better topical match to the query. Another example is the query “stars”.

Tf.Idf: List of The Simpsons episodes, List of stars on the Hollywood Walk of Fame, Star Wars, Star Trek, List of stars by constellation, Star, Star Trek Other Storylines

Google: Star, Neutron star, Flag of the United States, Movie star, List of nearest stars, Stars and Stripes, List of brightest stars

Topical filtering: Star system, Star (glyph), Star Trek Further Reading, Star (disambiguation), Star Wreck, Star, List of LucasArts Star Wars games

In this case, “Star (glyph)” is the mathematical concept of a star. In this case, the disambiguation page is only seen in the results from topical filtering, as well as a broader range of topical versions of star.

This paper first presents the background on discrete PCA (DPCA), and topic specific ranking using a topically motivated random surfer. Then the combination of these methods is described. The paper described the results of the topic specific ranking, a very appealing and rational set of document rankings for different topics. Finally the application of these techniques to information retrieval are discussed and presented.

Background

Topic Specific Ranking

We use the term “random surfer model” in a broad sense: to encompass general Monte Carlo Markov chain methods, modelling eye-balls on pages, used to determine scores for documents. Examples are (Haveliwala 2002; Richardson & Domingos 2002). A general method for topic-specific ranking roughly following ((Richardson & Domingos 2002) goes as follows:

Our surfer restarts with probability α at a page i with probability r_i . From that page, they uniformly select a link to document i' , and jump to this next page. They then consider the topic of the new page, whose strength of relevance is determined by another probability $t_{i'}$. With this probability $t_{i'}$ they accept the new page, and with probability $1 - t_{i'}$ they go back to the page i to try a new link. The stationary distribution of the Markov Chain for the probability of being on page p_i is then given by the update equations:

$$p_i \leftarrow \alpha r_i + (1 - \alpha) \sum_{i': i' \rightarrow i} p_{i'} \frac{t_i}{\sum_{j: i' \rightarrow j} t_j}$$

where we perform the calculation only for those pages i with $r_i > 0$, and $i' \rightarrow i$ denotes page i' links to page i . The vectors \vec{r} and \vec{t} allow specialization to a topic, so a set of such

¹<http://en.wikipedia.org>

²This link may not be functioning after 2005.

rankings \vec{p} can be developed for every topic: \vec{r} represents the starting documents for a topic and \vec{t} represents the probability that someone interested in the topic will stay at a page.

Note that previous applications of this technique have been hampered because full multi-faceted topical assignments for documents have not been available. Hence we apply discrete PCA to obtain a rich set of multi-faceted topics.

Discrete PCA

Principal component analysis (PCA), latent semantic indexing (LSI), and independent component analysis (ICA) are key methods in the statistical engineering toolbox. They have a long history and are used in many different ways. A fairly recent innovation here is discrete versions: genotype inference using admixtures (Pritchard, Stephens, & Donnelly 2000), probabilistic latent semantic indexing (Hofmann 1999) latent Dirichlet allocation (Blei, Ng, & Jordan 2003), discrete PCA (Buntine & Jakulin 2004) and Gamma-Poisson (GaP) models (Canny 2004) are just a few of the known versions. These methods are variations of one another, ignoring statistical methodology and notation, and form a discrete version of ICA (Buntine & Jakulin 2004; Canny 2004).

Each document is represented as an integer vector, \vec{w} , usually sparse. The vector may be as simple as bag of words, or it may be more complex, separate bags for title, abstract and content, separate bags for nouns and verbs, etc. The model also assigns a set of independent components to a document somehow representing the topical content. In the general Gamma-Poisson (GaP) model (Canny 2004) the k -th component is a Gamma(α_k, β_k) variable. In multinomial PCA or LDA it is a Gamma($\alpha_k, 1$) variable, but then the set of variables is also normalized to yield a Dirichlet (Buntine & Jakulin 2004). Finally, component distributions complete the model: each component k has proportion vector $\vec{\Omega}_k$ giving the proportion of each word/lexeme in the vector \vec{w} , where $\sum_j \Omega_{j,k} = 1$. The distribution for document \vec{w} , is then given using hidden components \vec{m} and model parameters $\vec{\Omega}$:

$$m_k \sim \text{Gamma}(\alpha_k, \beta_k) \quad \text{for } k = 1, \dots, K$$

$$w_j \sim \text{Poisson} \left(\sum_k \Omega_{j,k} m_k \right) \quad \text{for } j = 1, \dots, J$$

Alternatively, the distribution on \vec{w} can be represented using the total count of \vec{w} , $w_0 = \sum_k w_k$, as:

$$w_0 \sim \text{Poisson} \left(\sum_k m_k \right)$$

$$\vec{w} \sim \text{multinomial} \left(\sum_k \frac{\vec{\Omega}_k m_k}{\sum_k m_k}, w_0 \right)$$

If $\beta_k = \beta$ is constant as in LDA then this normalized \vec{m} is a Dirichlet and the totals safely ignored.

The family of models can be fit using mean field, maximum likelihood, Gibbs estimation, or Gibbs estimation using Rao-Blackwellization (Buntine 2005). These introduce

an additional set of quantities which are the word/lexeme counts w_j broken out into a term for each component, $w_{j,k}$ where $\sum_k w_{j,k} = w_j$ and $w_{j,k} \sim \text{Poisson}(\Omega_{j,k} m_k)$.

Setting up Topic Specific Ranking

Topic specific page rank can work off the normalized component values $m_k^* = m_k / \sum_k m_k$ for each document. For documents $i = 1, \dots, I$, let these be $m_{i,k}^*$. The restart vector \vec{r} for topic k can be given by $r_i = m_{i,k}^* / \sum_i m_{i,k}^*$. The topic relevance is more complicated. In general in discrete PCA, most pages may have a mix of topics with perhaps 5-10 different topics or components occurring for one document. Thus a document with $m_k^* = 0.2$ in these cases can be said to have the relevant topical content, since we rarely expect much more than 0.2. Thus, to derive the page relevance vector \vec{t} from discrete PCA, we put the $m_{i,k}^*$ through a scaled tanh function so that when $m_{i,k}^* = 0.2$, t_i will already be near 1.

Experiments: Sample Rankings

We downloaded the Wikipedia in April 2005. It has approximately 513,000 documents with over 2.5Gb of text, and a rich link structure. The lexicon of the top 310,000 nouns, 13,000 verbs, 31,000 adjectives and 3,500 adverbs are used in training. Words with less than 4 occurrences in the corpus are ignored. Words are stemmed and sorted this way because it greatly improves interpretability of the model.

We ran discrete PCA using Pritchard *et al.*'s Gibbs algorithm (Buntine 2005). with $K = 100$ components with Gamma(1/50, 1) priors, and using Jeffreys' prior for the component proportions Ω_k (Dirichlet with a constant vector of 1/2 for the parameters). This uses the MPCA software³, using a 600 cycle burn-in and 200 recording cycles, about 34 hours on a dual 3GHz CPU under Linux. Note that this sized corpus could easily support upto $K = 1000$ component model, but in this experiment we have chosen to limit the complexity of the search engine. Computing the set of 100 topic specific ranks for the documents takes 20 minutes using a naive algorithm with no handling of sparsity.

We compared some typical URLs (those with a high topic proportion) with those having a high rank for the topic in Table 1. A complete set of results for all components on this experiment can be viewed at our website⁴. Each topic has its own web page, accessed by clicking on the numbers, and document topic-specific rankings are given at the bottom of these pages. The difference between the typical titles (those for documents with a high topic proportion) and high-ranked titles is stark. High-ranked titles clearly describe the topic. Typical titles just give examples. For this reason, we believed that these topic-specific rankings could be used effectively in a search engine.

³Published at <http://cosco.hiit.fi/search/MPCA>

⁴See the *topic browser* at the demonstration Wikipedia search engine.

Common nouns	Typical titles	High-ranked titles
Star, Earth, Moon, Sun, planet, objects, astronomer, Galaxy, asteroids	204 Kallisto, 217 Eudora, 228 Agathe, 266 Aline, 245 Vera, 258 Tyche, 219 Thusnelda	Astronomy, Earth, Sun, Moon, Star, Asteroid, Astronomer
language, word, English, IPA, name, Unicode, dialect, letter, span	List of consonants, List of phonetics topics, Digraph (orthography), Table of consonants, Ubykh phonology, Code page 855,	IPA chart for English English language, Language, Latin, Linguistics, Greek language, French language, International Phonetic Alphabet
theory, term, example, people, philosophy, time, idea, work, World	Incommensurability, Qualitative psychological research, Social constructionism, Culture theory, Internalism and Externalism, Ethical egoism, Positive (social sciences)	Philosophy, Psychology, Mathematics, Economics, Science, Biology, Physics
music, composer, instruments, opera, song, piano, Orchestra, work, Symphony	Piano quintet, String quintet, List of atonal pieces, List of pieces which use the whole tone scale, String trio, Piano sextet, Trio sonata	Music, Composer, Opera, Musical instrument, Classical music, Jazz, Piano
mythology, God, goddess, son, deities, Greek mythology, Norse, name, myth	Tethys (mythology), Uranus (mythology), Oceanid, Psamathe, Phorcys, List of Greek mythological characters, Galatea (mythology)	Greek mythology, Mythology, Norse mythology, Polynesian mythology, Roman mythology, Zeus, Homer

Table 1: A sample of components

Using Discrete PCA in Information Retrieval

PLSI introduced by (Hofmann 1999) was first suggested as an approach for information retrieval, and the GaP model has also been applied here by (Canny 2004). The general method for applying it is the so-called language modelling approach to information retrieval of (Ponte & Croft 1998). This goes as follows: one develops a statistical model for each document, denote the model for the i -th document by \mathcal{D}_i . Under this model, one can pose questions such as, what is the probability that query words \vec{q} would also be added to the document? This is $p(\vec{q} | \mathcal{D}_i, \mathcal{M})$. where the model construct \mathcal{M} specifies the form used. This approach then looks to retrieve the document i maximising this probability.

The effort then is placed in the development of the so-called language models which are depend on individual documents \mathcal{D}_i . This needs to be a very flexible model because it needs to work for any smaller query set \vec{q} . (Azzopardi, Girolami, & van Risjbergen 2003) have shown that high perplexity general models, ones with high values for $p(\mathcal{D}_i | \mathcal{M})$, are not always useful for information retrieval. We conjecture that a significant part of this may be that high perplexity models are not necessarily good at predicting individual words. That is, while the quality of $p(\mathcal{D}_i | \mathcal{M})$ can be good, and experiments show this is the case for discrete PCA (Hofmann 1999), it does not imply that the quality of $p(\vec{q} | \mathcal{D}_i, \mathcal{M})$ will follow. Information retrieval applied to a large news corpus should really build a model relevant to the two words “german immigration”, or another two words “molecular biology”, not to the whole corpus in one go. The minimum description length intuition is that bits used to describe the general model are wasted for the specific task.

Traditionally, language modeling has achieved reasonable performance by a compromise. The probability of a word

q_j in a query is usually obtained by smoothing the model probability $p(q_j | \mathcal{D}_i, \mathcal{M})$ with the observed frequency of the word in the document itself. Suppose the frequency of the word q_j in the i -th document is $\hat{p}(q_j | \vec{w}_i)$, then use the probability

$$\alpha p(q_j | \mathcal{D}_i, \mathcal{M}) + (1 - \alpha) \hat{p}(q_j | \vec{w}_i).$$

This trick has allowed the method to achieve impressive results in some applications such as web search where separate models for title words, link text, etc. were combined by (Craswell & Hawking 2003). It is not clear at this stage, however, whether this trick represents some some fundamental theoretical property or correction term of language modelling for information retrieval.

When a high perplexity discrete PCA model is applied without this smoothing, performance is not always good, but if the query is rather general, it can be surprisingly good. Some examples are presented by (Buntine, Perttu, & Tuulos 2004; Buntine & Jakulin 2004). Intuitively, for general queries where $p(\vec{q} | \mathcal{D}_i, \mathcal{M})$ has significant statistical support from the model $p(\mathcal{D}_i | \mathcal{M})$, better performance in information retrieval might be expected. Thus one approach to using discrete PCA in information retrieval is to use query probabilities as a way of scoring broad topical relevance of a document, and thus combining it with other retrieval scores. That is, apply discrete PCA in situations where we expect the high perplexity model to translate to a high fidelity query probability $p(\vec{q} | \mathcal{D}_i, \mathcal{M})$.

Information Retrieval with Topic Specific Ranking

Taking the previously discussed experience and views into consideration, we developed a search engine that uses stan-

dard Tf.Idf as its retrieval engine, and then does post-filtering (i.e., re-ordering) or retrieved documents using topic specific page rank. We use the Okapi BM25 version of Tf.Idf described in (Zhai 2001), recoded within our indexing system. The top 500 documents with no less that 25% of the Tf.Idf score of the best document are retained from a query q and put through the reranking phase.

For the query q , we also have topic words t that may be the same as q (if obtained from our initial search screen) or may be different (if obtained from subsequent search screens). For the query words t , the normalized component proportions (see section on discrete PCA) are estimated using Gibbs importance sampling with 2000 cycles (Buntine & Jakulin 2004), to yield the 100-dimensional normalised vector \vec{m}_t^* . A topic-specific ranking probability is then obtained for each page i by making then linear product of \vec{m}_t^* with the $K = 100$ topic specific page ranks for the page represented as a 100-dimensional vector \vec{r}_i . This is then combined with the Tf.Idf score to produce a final ranking for the i -th document:

$$C * Tf.Idf(q, i) + \log \left(\sum_k r_{i,k} m_{t,k}^* \right) \quad (1)$$

This heuristic formula is justified as follows:

- while Tf.Idf is not properly calibrated to any probability, we guess it is best viewed as a log probability, but of unknown scale⁵,
- the constant C with we currently set to 0.05 is then intended to convert it to units of log probability,
- the sum inside the log is our approximation to what the topic specific page rank for topic words t would be for each page.

This formula is only evaluated on at most 500 documents, so is relatively cheap to do.

The performance of this technique can be evaluated by using the search engine demonstrated at our test website. The commentary pages at the site also give details of the results of the topic-specific link analysis performed here. To view results with Tf.Idf alone, after the first query is done, blank the content of the “context words” box and resubmit a query.

Examples of Queries

We briefly present here a number of examples. For the query “grand piano keys”, the top results by either method (Tf.Idf or topical filtering) are:

Piano, Grand piano, Piano acoustics, Player piano, Instant Karma! Innovations in the piano

The major results of the query have a consistent topical content so the methods do not vary. This is the most common result when using the search engine with context words the same as the query. For the query “jazz musician playing clarinet,” topical filtering yields (taking context words from the query)

Ted Lewis (musician), Pee Wee Russell, Benny Goodman, Dixieland, Han Bennink, Louis Armstrong and his Hot Five, Leon Roppolo

and Tf.Idf yields

Dixieland, Music of the United States before 1900, Benny Goodman, Music of Brittany, Pee Wee Russell, Klezmer, List of jazz musicians.

The latter has more irrelevant entries.

This next example illustrates biasing the search with different context words. For the query “madonna” with context words “paintings and statues”, topical filtering yields

The Madonna of Port Lligat, Black Madonna, Madonna and Child (Duccio), Pier Antonio Mezzastris, Madonna (art), The Madonna, Madonna Inn

and Tf.Idf with the query ‘madonna paintings and statues’ yields

Leonardo da Vinci, List of artwork, Michelangelo Buonarroti, Quito, Vizzini, Icon, List of statues on Charles Bridge

One sees a better emphasis in topical filtering on Madonna, whereas in Tf.Idf the topic words swamp the query.

Conclusion and Future Work

Topic specific scoring provided by the adapted random surfer model, as shown by the Wikipedia examples, provides a far more characteristic score for documents than the proportion of component. The titles of high-ranking documents are indicative of the component, and in many cases can serve as reasonable component titles or descriptions. In contrast, documents containing a large proportion of the component are best described as “typical”. They are neither indicative or characteristic. Topic-specific link analysis is therefore a valuable tool for the interpretation of topics developed by discrete PCA.

The ranking works well as a *topically biased post-ranking filter* for standard information retrieval. Experience on the Wikipedia search engine so-developed shows the resultant retrieval to be effective in many cases, though it has a small negative effect in a few cases. In more than half the cases, where there is no topical ambiguity, it appears no different to regular Tf.Idf. Preliminary testing shows the method has promise enough to justify a larger user study. Notice that for a comparative study, comparisons should be made with Google (i.e., using a web-based ranking scheme), and with Tf.Idf augmented like Equation (1) but using standard pagerank, without topic specialization.

Perhaps the best potential for the topically biased post-ranking filter, however, is that it provides an effective means for users to bias their search in topical directions. This ability is suggested by web commentary on search engines, and serves as a naive and immediately available counterpart to full semantic web capability, which is not currently available. While this has no counterpart in existing information retrieval, it is also something that needs to gain acceptance from the fickle users of search engines.

⁵Clearly questionable since it can also be viewed as a utility.

Acknowledgments The work was supported by the ALVIS project, funded by the IST Priority of the EU's 6th framework programme, and the Search-Ina-Box project, funded by the Finnish TEKES programme. It benefits greatly from discussions with Natalie Jhaveri and Tomi Heimonen and of the Tampere Unit for Computer-Human Interaction at University of Tampere.

References

- Azzopardi, L.; Girolami, M.; and van Risjbergen, K. 2003. Investigating the relationship between language model perplexity and IR precision-recall measures. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, 369–370.
- Blei, D.; Ng, A.; and Jordan, M. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.
- Buntine, W., and Jakulin, A. 2004. Applying discrete PCA in data analysis. In *UAI-2004*.
- Buntine, W.; Perttu, S.; and Tuulos, V. 2004. Using discrete PCA on web pages. In *Workshop on Statistical Approaches to Web Mining, SAWM'04*. At ECML 2004.
- Buntine, W. 2005. Discrete principal component analysis. submitted.
- Canny, J. 2004. GaP: a factor model for discrete data. In *SIGIR*, 122–129.
- Craswell, N., and Hawking, D. 2003. Overview of the TREC 2003 web track. In *Proc. TREC 2003*.
- Haveliwala, T. 2002. Topic-specific pagerank. In *11th World Wide Web*.
- Hofmann, T. 1999. Probabilistic latent semantic indexing. In *Research and Development in Information Retrieval*, 50–57.
- Johnson, S. 2003. Digging for googleholes. *Slate*. <http://slate.msn.com/id/2085668/index.html>.
- Nallapati, R. 2004. Discriminative models for information retrieval. In *ACM SIGIR Conference*.
- Ponte, J., and Croft, W. 1998. A language modeling approach to information retrieval. In *Research and Development in Information Retrieval*, 275–281.
- Pritchard, J.; Stephens, M.; and Donnelly, P. 2000. Inference of population structure using multilocus genotype data. *Genetics* 155:945–959.
- Richardson, M., and Domingos, P. 2002. The intelligent surfer: Probabilistic combination of link and content information in pagerank. In *NIPS*14*.
- Suomela, S., and Kekäläinen, J. 2005. Ontology as a search-tool: A study of real users' query formulation with and without conceptual support. In *ECIR 2005*, 315–329.
- Zhai, C. 2001. Notes on the Lemur TFIDF model. Note with Lemur 1.9 documentation, School of CS, CMU.