Supporting the Exploration of a Corpus of 17th-Century Scholarly Correspondences by Topic Modeling

Peter Wittek **Walter Ravenek**
Huygens Institute for the History of the Netherlands
Prins Willem-Alexanderhof 5, 2595 BE Den Haag
The Netherlands

Abstract

This paper deals with the application of topic modeling to a corpus of 17th-century scholarly correspondences built up by the CKCC project. The topic modeling approaches considered are latent Dirichlet allocation (LDA), latent semantic analysis (LSA), and random indexing (RI). After describing the corpus and the topic modeling approaches, we present an experiment for the quantitative evaluation of the performance of the various topic modeling approaches in reproducing human-labeled words in a subset of the corpus. In our experiments random indexing shows the best performance, with scope for further improvement. Next we discuss the role of topic modeling in the *CKCC Epistolarium*, the virtual research environment that is being developed for exploring and analysing the CKCC corpus. The key feature of topic modeling is its ability to calculate similarities between words and texts. In an example we illustrate how such an approach may yield results that transcend a regular text search.

1 Introduction

As part of the CKCC project we are investigating topic modeling and analysis of topic dynamics in a multilingual corpus of approximately 19,000 letters. CKCC is an acronym for *Circulation of Knowledge and Learned Practices in the 17th-Century Dutch Republic. A Web-based Humanities Collaboratory on Correspondences* (Roorda et al., 2010), a project of a Dutch consortium of universities, research institutes and cultural heritage institutions, collaborating to provide tools for analyzing a machine-readable and growing corpus of letters of scholars exchanging information in the 17th century. The central research question is how to combine letter texts and metadata in such a way that we can analyze the circulation and appropriation of knowledge production in a wider international context and recognize the development of themes of interest and scholarly debates in space and time.

2 The Corpus

The CKCC corpus currently contains 19,239 letters. It consists of correspondences of Caspar Barlaeus (1584-1648), Isaac Beeckman (1588-1637), Hugo de Groot (1583-1645), Constantijn Huygens (1596-1687), Christiaan Huygens (1629-1695), and Antoni van Leeuwenhoek (1632-1723). Three other correspondences, among which that of René Descartes (1596-1650), will be added in the near future.

From a language perspective our corpus has a number of characteristics that are important when it comes to processing the letters:

- The corpus contains letters in various different languages, the most important ones being Dutch, French and Latin. As can be seen from Table 1, these three languages account for about 95% of the text.

- The letters are not monolingual: in many letters various languages are used alternatingly. In order to apply language resources and technology we have to segment the letters to at least paragraph level. Currently we are using an N-gram based language identification algorithm (Ahmed et al., 2004) for assigning languages to each paragraph. The language profiles are constructed using a selected set of letters from the corpus.

- The letters often contain elaborate opening and closing phrases that contribute little to the subject matter of the letters. It may be worthwile to exclude such phrases from content extraction.

- Finally, 17th-century writing contains a large spelling variation. For instance, in our corpus the name *Christiaan Huygens van Zuylichem* is spelled in at least 320 different ways.

language	paragraphs	tokens	rel. size
Dutch	44,680	2,427,805	33.1%
English	1,942	89,759	1.2%
French	27,197	2,219,862	30.3%
German	4,634	116,405	1.6%
Greek	7	91	0.0%
Italian	5,052	76,205	1.0%
Latin	37,798	2,254,556	30.7%
Not Assigned	15,199	150,472	2.1%
Total	136,509	7,335,155	

Table 1: Corpus size by language.

3 Topic Modeling

Topic modeling constitutes a statistical approach to content extraction. The major approaches to topic modeling are able to identify hidden variables that can be interpreted as 'topics'. We consider the following three methods: latent Dirichlet allocation (LDA), latent semantic analysis (LSA), and random indexing (RI). Each of these approaches is derived from the so-called vector space model. Intuitively, if text fragments of two documents address similar topics, it is highly possible that they share many substantive terms. Conversely, if two terms occur in many documents together, the terms are likely to be related.

After preprocessing the letters, we construct a vector representation for each document. Let $\mathbf{a}_j$ be a document vector in the vector space model, that is, $\mathbf{a}_j = \sum_{k=1}^{M} a_{kj}\mathbf{e}_k$, where M is the number of index terms, a_{kj} is some weighting (e.g., term frequency), and the vectors $\mathbf{e}_k$ form a basis for the M-dimensional Euclidean space. The matrix A with elements a_{kj} is called the co-occurrence matrix; its rows correspond to term vectors.

Given this representation, semantic relatedness of a pair of text fragments is computed as the cosine similarity of their corresponding term vectors which is defined as

$$S(\mathbf{a}_i, \mathbf{a}_j) = \frac{\mathbf{a}_i \cdot \mathbf{a}_j}{|\mathbf{a}_i||\mathbf{a}_j|}.$$

3.1 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA) uses a generative probabilistic model of a corpus. The basic idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. LDA assumes the following generative process for each document d in a corpus D (Blei et al., 2003):

1. Choose $\theta \sim \mathrm{Dir}(\alpha)$.
2. For each of the N terms t_n:
   (a) Choose a topic $z_n \sim \mathrm{Multinomial}(\theta)$.
   (b) Choose a term t_n from $p(t_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

Given the above generative process, the task is to compute the posterior distribution of the hidden variables given a document:

$$p(\theta, z|d, \alpha, \beta) = \frac{p(\theta, z, d|\alpha, \beta)}{p(d|\alpha, \beta)}.$$

While this formula is computationally intractable in most cases, approximations exist.

3.2 Latent Semantic Analysis

Conceptually, latent semantic analysis (LSA), or latent semantic indexing, is similar to the generalized vector space model (Wong et al., 1985). LSA treats the entire document as the context of a word being analyzed. In LSA, the dimension of the vector space is reduced by singular value decomposition (Deerwester et al., 1990). The singular values of A are gained by the eigen base of A. Let U denote the matrix of left singular vectors, and V the matrix of right singular vectors. Let Σ denote a rectangular matrix, its diagonal consisting of the singular values, the other elements are zero. By the orthogonality of U and V, the following decomposition is derived: $A = U\Sigma V^*$. This formula is the singular value decomposition of the matrix A. Let Σ_k denote that matrix which is similar to Σ, but it has only the k highest singular values in its diagonal. Then $A_k = U\Sigma_k V^*$, and A_k is the best approximation to A for any unitarily invariant norm (Berry et al., 1995; Mirsky, 1960).

Using rank reduction to get the so-called feature space, terms that occur together very often in the same documents are merged into a single dimension of the feature space, and these merged features will be the topics to be modeled. The dimensions of the reduced space correspond to the axes of greatest variance. The number of topics is hinted by the singular values: if there is a great drop in consecutive values, that can be regarded is a cut-off point. On large English corpora, this normally occurs between two and five hundred topics (Bradford, 2008).

3.3 Random Indexing

Random indexing (RI), or random projection, does not rely on computationally intensive matrix decomposition algorithms like singular value decomposition. Instead of first constructing the co-occurrence matrix A and then using a separate dimension reduction phase, RI builds an incremental word space model (Kanerva et al., 2000; Sahlgren, 2005). The random indexing technique can be described as a two-step operation:

- First, each context (e.g., each document or each word) in the data is assigned a unique and randomly generated representation called an index vector. These index vectors are sparse; they consist of a small number of randomly distributed $+1$s and -1s, with the rest of the elements of the vectors set to 0.
- Then, context vectors are produced by scanning the text, and each time a word occurs in a context (e.g., in a document or in a sliding context window), that context's index vector is added to the context vector for the word in question. Words are thus represented by context vectors that are effectively the sum of the words' contexts.

Random indexing does not provide an explicit way of computing the number of topics, it is a parameter of the model. The possibility to use a sliding window allows proximity of words to be taken into account in the topic modeling, which differentiates RI from LDA and LSA.

4 Evaluating Modeling Approaches

In this section we describe our approach to evaluation of topic modeling approaches and present results obtained so far.

4.1 Approach

In order to evaluate the various topic modeling approaches in the context of our 17th-century letter corpus, we use an approach that regards topic modeling as a variant of text classification, where the classes are topics, and a letter may belong to any number of classes. To calculate precision and recall with regard to a class c_k, estimates can be obtained as (Sebastiani, 2002):

$$\text{precision}(c_k) = \frac{TP_k}{TP_k + FP_k},$$

$$\text{recall}(c_k) = \frac{TP_k}{TP_k + FN_k},$$

where TP_k is the number of correctly classified instances under c_k, FP_k is the number of false positives, and FN_k is the number of false negatives, that is, the errors of omission. Precision and recall should be interpreted together, they are not sensible measures of effectiveness in themselves. It is well known from information retrieval practice that higher levels of precision may be obtained at the price of lower values of recall (van Rijsbergen, 1979). Therefore a classifier should be evaluated using a measure that combines precision and recall. We used the widely adopted F_1 function, which is the harmonic mean of recall and precision:

$$F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

The above measures assume that we know the categories, we know which documents contain which topics. In other words, that we can tell which are the true positives and false positives. This is not true in the case of the CKCC corpus, and it is not feasible to label the entire corpus by experts. Benchmarks, however, are still possible if one considers a subset of the letters. To enable a meaningful comparison of methods, we adopted the following procedure:

1. We randomly selected a subset of a three hundred letters, hundred for each of the three major languages – Dutch, French, and Latin.

2. We asked experts to annotate the selected letters by labeling about twenty words that represent the topics discussed in the letter. The number was not fixed because the letters have substantially different lengths.

3. We extracted topic labels for each letter with each of the modeling methods and compared the result with the annotation to obtain an F_1 score.

Given the above annotated subset, we were able to automate the benchmarking process, and select the best setting of parameters. We benchmarked the impact of removal of stop words, stemming, and spelling correction, as well as the three modeling alternatives. We also compared two methods for scoring similarity, one based on the similarity measure described in Section 3, and another one that sorts the results by term frequency, putting the most frequent terms first.

4.2 Results

In Table 2 we present the main results for comparison of the topic modeling approaches and the language technologies used. Having studied the impact of the number of dimensions, we found that none of the three methods is sensitive to this parameter. We settled with two hundred dimensions, as it gave a good performance and it was computationally effective. We also favored the similarity measure as described in Section 3. We extracted 10, 20, ..., 100 topic labels, and averaged the F_1 values to obtain a single score for each experiment. We restricted the vocabulary of the extracted labels to the words that occur in the document, since the experts who labeled the documents had also been asked to do so.

Experiment	LDA	LSA	RI
fr	0.0633	0.0970	0.0565
fr+stop	0.1076	0.1102	0.1564
fr+stop+stem	0.1027	0.0908	0.1477
fr+stop+spell	0.0804	0.0509	0.0925
fr+stop+spell+stem	0.0872	0.0759	0.1086
la	0.0695	0.1327	0.0616
la+stop	0.1370	0.1204	0.1753
nl	0.0841	0.1626	0.1009
nl+stop	0.1249	0.1605	0.2435
nl+stop+stem	0.1575	0.1582	0.2482

Table 2: F_1 for various experiments.

The experiments are characterized as follows: 'fr' indicates a run on French letter texts with no preprocessing except tokenization and lowercasing, 'fr+stop' indices an experiment with additional stop word removal, 'fr+stop+stem' indicates an experiment with additional stemming applied; finally, the label 'spell' is used to indicate the use of spelling correction. Note that we did not manage to get a Latin stemmer that functioned properly in the Lucene framework employed by us.

4.3 Discussion

We use the 'fr+stop', 'la+stop', and 'nl+stop' experiments for comparison of the performance of the topic modeling approaches. From our results it can be seen that random indexing performs best, whereas there is not much difference between latent Dirichlet allocation and latent semantic analysis. It is to be noted that our current RI implementation uses the complete document as window, so proximity effects have not yet

been investigated. This offers scope for further improvement of the RI results.

With respect to the effect of using language technologies we note that stop word removal has a significant effect, whereas the effect of stemming is not so large. Also, random indexing appears to be less sensitive to the use of language technologies.

We performed some initial experiments for investigating the effect of spelling correction for French texts. We employed the VARD 2 application (Baron and Rayson, 2008), which has been successfully used for dealing with spelling variation in Early Modern English texts. As can be seen from Table 2, including the spelling correction deteriorates the results. More research is needed to explain this behavior, but we can think of a number of reasons: The labeling of the letters was done by different researchers, each having his own language; we are currently analyzing the consistency of the labeling effort. Secondly, we perform language identification on a paragraph basis. In many letters, especially in the Constantijn Huygens corpus, French and Dutch are often used in the same paragraph. It seems that in our current experiments many Dutch words are 'pulled' into the French domain, which is likely to have a negative effect.

5 Leveraging on Topic Modeling

The preceding section deals with the evaluation of various topic modeling approaches in terms of a well-defined framework. In the end, the usefulness of topic modeling lies in its capabilities to support the user in exploring the corpus in ways that transcend a standard full text search. The key feature in this respect is the possibility to calculate similarities between words and documents.

We see two major applications:

- Enhance the full text search by including terms suggested by the topic model;
- Allow the user to find documents that have largest similarity with a given text fragment.

Maybe the last application comes closest to 'asking the unaskable': it does not require specification of search terms, but merely a selection of text that interests the user. Such a text fragment contains words that could be entered in the full text search, but also less-obvious words that may contribute to meaning in more subtle ways.

5.1 Implementation

We are currently developing a virtual research environment, the *CKCC Epistolarium*, that provides for browsing and analysing the letters in our corpus. It allows the user full text searches in combination with a selection based on the metadata of the letters (date, sender, recipient, location of sending, location of receipt). The *Epistolarium* builds on the Lucene text search library (Gospodnetic et al., 2005). Lucene has an open architecture and comes with a wide array of text analyzers and filters; if needed, one can easily develop and include custom components. In our topic modeling experiments we have used *Semantic Vectors* (Widdows and Ferraro, 2008) for LSA and RI, and the machine learning toolkit *Mallet* (McCallum, 2002) for LDA. *Semantic Vectors* builds on the indexes created by Lucene, which makes it very attractive from an architectural point of view. Since the results of our experiments indicate that LSA and LDA perform equally well on our corpus, we decided to use *Semantic Vectors* and to focus on LSA and RI.

We have started implementation of the incorporation of topic modeling results in the *Epistolarium*. Figure 1 shows the architecture for indexing and model building. We have implemented two parallel pipelines: one language-agnostic and one language-specific. Each document in the corpus is processed by both pipelines (see the leftmost side of Figure 1). A simple preprocessor deals with words and characters that can be removed confidently irrespective of language (Figure 1, left side, upper part); this includes numbers, Roman numerals, words shorter than 3 characters, etc. The language specific preprocessor currently deals with stop words and stemming. It also includes a spelling correction module that was built with VARD (Baron and Rayson, 2008). The preprocessor may be extended in future work (Figure 1, left side, lower part). For n languages there are $n+1$ inverted Lucene indices: one for each language using the language-specific preprocessor, and one index of the entire corpus using the language-agnostic preprocessor. All indices are channeled to the two modeling methods (Figure 1, right side).

Figure 2 shows the proposed architecture for retrieval of suggested search terms and similar documents. In the text search the user can request additional query terms based on similarity with user-specified terms. The query terms are handled according to the language preferences (see the left side of Figure 2). The query terms are forwarded to the respective topic models. The LSA or RI module returns a ranked list of keywords which are the most relevant to the topic or topics underlying the query (Figure 2, right side). The user can refine his or her search based on the suggestions.

The other usage scenario works in a similar fashion. The user has the option to specify a text fragment of interest in one of the letters, found by previous browsing or searching the corpus. Again, the text fragment is processed according to the language preferences and the LSA or RI module returns a ranked list of document which are the most relevant to the topic or topics underlying the input (Figure 2, right side).

5.2 Example

We have performed a similarity calculation for a number of text fragments. One of these reads, translated

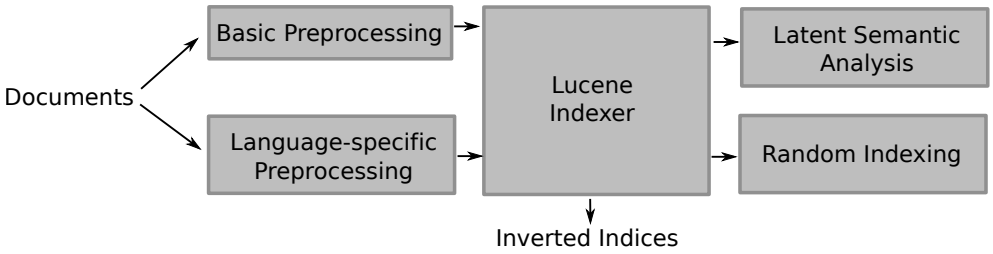

Figure 1: Indexing and model building.

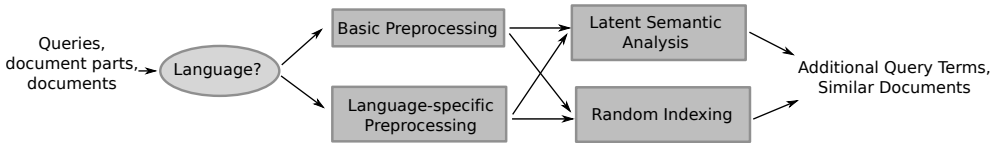

Figure 2: Retrieval of query terms and similar documents.

from Dutch into English:

> *Beauregard starts a movement to ask the King of England to appoint a catholic governor, and has called a meeting for this purpose. The King has been written. What can we do against this?*

which occurs in a letter (huygens/6125) dated July 15, 1663 from Gaspard de la Pize to Constantijn Huygens, acting as a representative of the house of Orange to regain control over the princedom Orange in southern France.

Our software returns a list of documents, the first five of which are (with cosine similarity in parenthesis):

> huygens/6125 (0.904)
> huygens/6124 (0.630)
> huygens/6250 (0.547)
> huygens/5953 (0.530)
> huygens/5867 (0.524)

First of all we notice that letter huygens/6125 is at the top of our list. On inspection, the second and fifth letters deal with the same problem as described in the text fragment. The third letter is somewhat unclear to us, but the fourth one, huygens/5953, is quite interesting: it shares only Beauregard as an obvious keyword, but it certainly deals with trouble stirred up by him in Orange.

Next, we performed a regular full text search with terms specified in the text fragment (we choose the Dutch equivalents of Beauregard, England, catholic and governor). The top five results, ordered by relevance, are:

> huygens/6125
> huygens/6124
> huygens/5867
> huygens/5839
> huygens/6004

Again huygens/6125 is at the top of the list. On inspection we find that all letters are relevant for the issue addressed in the text fragment we started with. However, none of them goes beyond the search terms specified, as is the case with letter huygens/5953 obtained with topic modeling.

We think that this example illustrates the potential usefulness of topic modeling in the exploration of the letter corpus: it may direct the user to letters that may be hard to find with a regular text search.

6 Conclusions

We have investigated the performance of various topic modeling approaches on a multilingual corpus of 17th-century correspondences. An evaluation based on comparison of results with a subset human-labeled letters indicates that random indexing performs best, whereas the difference between latent Dirichlet allocation and latent semantic indexing is small. Results for RI may improve if we take word proximity into account. Language technologies can yield improvements in the results, but considerable care has to be taken in their application.

Topic modeling can be used in the exploration of the corpus by enhancing the full text search with query terms suggested by the topic model. Furthermore, the user can request letters that have the largest similarity with an arbitrary selected text fragment.

7 Acknowledgements

We thank Frans Blom, Erik-Jan Bos, Eric Jorink, Dirk van Miert, Henk Nellen, and Huib Zuidervaart for the labeling of the set of 300 letters. We thank Alistair Baron for providing the VARD 2 tool, extended with an interface that makes integration with our software very convenient. Erik-Jan Bos performed the training

of VARD 2 for French. Finally, we thank Charles van den Heuvel for critically reading the manuscript.

The CKCC project has funding of the Netherlands Organization of Scientific Research (NWO). Furthermore it received a grant of CLARIN-NL to acquire language technology expertise, which was used to hire the first author.

References

Ahmed, B., Cha, S., and Tappert, C. (2004). Language identification from text using n-gram based cumulative frequency addition. In *Proceedings of Student/Faculty Research Day, CSIS, Pace University*.

Baron, A. and Rayson, P. (2008). VARD 2: A tool for dealing with spelling variation in historical corpora. In *Proceedings of the Postgraduate Conference in Corpus Linguistics*, Birmingham, UK.

Berry, M., Dumais, S., and O'Brien, G. (1995). Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595.

Blei, D., Ng, A., and Jordan, M. (2003). Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022.

Bradford, R. (2008). An empirical study of required dimensionality for large-scale latent semantic indexing applications. In *Proceeding of CIKM-08, 17th Conference on Information and Knowledge Management*, pages 153–162, Napa Valley, CA, USA.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Gospodnetic, O., Hatcher, E., et al. (2005). *Lucene in Action*. Manning.

Kanerva, P., Kristofersson, J., and Holst, A. (2000). Random indexing of text samples for latent semantic analysis. In *Proceedings of CogSci-00, 22nd Annual Conference of the Cognitive Science Society*, volume 1036, Philadelphia, PA, USA.

McCallum, A. (2002). Mallet: A machine learning for language toolkit.

Mirsky, L. (1960). Symmetric gage functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11:50–59.

Roorda, D., Bos, E.-J., and van den Heuvel, C. (2010). Letters, ideas and information technology: Using digital corpora of letters to disclose the circulation of knowledge in the 17th century. In *Proceedings of Digital Humanitites*, London, UK.

Sahlgren, M. (2005). An introduction to random indexing. In *Proceedings of TKE-05, Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering*, Copenhagen, Denmark.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworths, London, UK.

Widdows, D. and Ferraro, K. (2008). Semantic vectors: a scalable open source package and online technology management application. In *LREC-08, 6th International Conference on Language Resources and Evaluation*, Marrakech, Morocco.

Wong, S., Ziarko, W., and Wong, P. (1985). Generalized vector space model in information retrieval. In *Proceedings of SIGIR-85, 8th International Conference on Research and Development in Information Retrieval*, pages 18–25, Montréal, Québec, Canada.