A text mining case study on document clustering and topic extraction

Introduction

Text mining can be defined as the analysis of large scale volumes of documents using statistical and machine learning methods. Development and application of text mining methods may significantly alleviate the resources and time that is otherwise spent in analysis. In addition the information that can be obtained using machine learning approaches is much richer and based on a more solid ground than what can be achieved with more conventional methods such as expert reviewing.

The initial steps in any text mining approach generally involve document parsing, filtering out semantically meaningless words and formatting to a matrix of document-terms vectors that can be used as input to different machine learning algorithms.
In the next steps, and since the document-term matrix can get very large, optional dimensionality reduction algorithms can be applied. This is the case for the method known as LSA (Latent Semantic Analysis), which uses a type of matrix decomposition (SVD, Singular Value Decomposition) to reduce the original matrix to smaller ones, in the documents and the terms vector spaces.

In the proposed use case, the corpora of sets of documents are already processed to the point of a bag-of-words per corpus. In this approach, a corpus of documents is represented by a file with the vocabulary of unique words, and a file with indexed absolute frequencies per document per term. While disregarding word order and context, this approach has been proven to be very useful for document clustering and topic extraction in many occasions.

Furthermore, the provided corpora are not labeled in any way. There is no previous knowledge on the number or the nature of the topics that there may be present.
In this sense an approach involving an initial step of LSA (Latent Semantic Analysis) performed via TruncatedSVD for dimensionality reduction followed by K-Means, using a range of possible cluster numbers, provides a pipeline that combines document clustering and topic extraction within each of the clusters independently.

Material and Methods

The analysis of this case study was done entirely using Python (v2.7.14) and libraries Scipy, Numpy, Pandas, Sklearn. For a complete list of the requirements see the accompanying jupyter notebook https://github.com/vitalv/doc-clustering-topic-modeling/blob/master/doc-clustering-topic-modeling.ipynb For the largest datasets the distributed computing framework Spark was used

Description of the datasets

The analyzed datasets in this study consist of a set of text corpora that have already been preprocessed to a bag-of-words format and cover a large range in terms of size (from 1500 documents in the smaller corpus to 8 200 000 in the larger) and vocabulary (minimum vocabulary size is 6906, maximum is 141043). The documents have no classes or metadata whatsoever, except for an identfier and a vague description. From smaller to larger in number of documents the datasets are the following: ‘KOS’ comprised blog entries; ‘NIPS’, texts from a conference on Neural Processing Systems; ‘ENRON’, a collection of e-mails , ‘NYTIMES’, articles from the New York Times newspaper, and ‘PUBMED’, biomedical publications from the american National Center for Biotechnology Information.

Generating and weighting Document-Term matrices

For every pair of vocabulary and frequencies files, custom functions were implemented to read in the data, generating a list for the vocabulary and a *compressed sparse row* Document-Term matrix. Next, absolute term frequencies (word counts) were transformed into the more robust metric TFIDF (*Term Frequency Inverse Document Frequency*). By introducing into the equation the frequency of each term across the corpus and the length of each document, TFIDF is a more reliable measure of how discriminative words are as topic representing. In addition, it might help improve the performance in clustering and class prediction.

Equation 1) $$idf_t = \log_2\left(\frac{N}{df_t}\right)$$

Equation 2) $$tf\,idf_t = tf_{(t,d)} \times idf_t$$

Singular Value Decomposition (SVD) of the Document-Term matrix

The DTM is very sparse. For a particular document (row), out of all the terms in the vocabulary, only a few of them will be nonzero in the vector. One important implication is that there are columns in which most values will be zero, representing almost meaningless terms that can be regarded as noise. A common approach to get rid of this useless columns is to perform some dimensionality reduction to the weighted DTM.
TruncatedSVD is a popular choice in this sense. When applied to document-term matrices it is also known as Latent Semantic Analysis (LSA) because it can be seen as a way to transform the matrix into a lower dimensional “semantic” space.
For a DTM X, the approximated X’ SVD decomposition is:

Equation 3) $$X' \approx U\,D\,V^T$$

LSA requires an user-defined value of *s* singular values. The smaller, the greater dimensionality reduction, at the cost of losing part of the structure in the original matrix.
In the current study, values of *s* where independently chosen for every corpus of documents such that at least 75% of the original variance was preserved in the dimension reduced DTM selected to preserve as much of the variance structure

K-Means clustering

The K-Means algorithm fundamental concept is to separate samples in groups of equal variance. In the first step it chooses the initial centroids (k, the number of clusters) and next, after initialization, the algorithm loops between the two other steps:
The first step assigns each sample to its nearest centroid. The second step creates new centroids by taking the mean value of all of the samples assigned to each previous centroid and such that the inertia or within-cluster sum-of-squares is minimized.

Results

The main components in the applied text mining pipeline consisted of a dimensionality reduction via LSA, followed by the K-Means clustering algorithm . In all the studied datasets an optimal number of clusters could not be estimated (Figure 1) indicating very interrelated topics in the datasets. Different numbers of clusters will generate different levels of granularity in the extracted topics. In this sense, a low number of clusters will provide broadly defined topics as depicted in Figure 2a whereas a higher number of clusters enables a more fine grained categorization (Figure 2b)

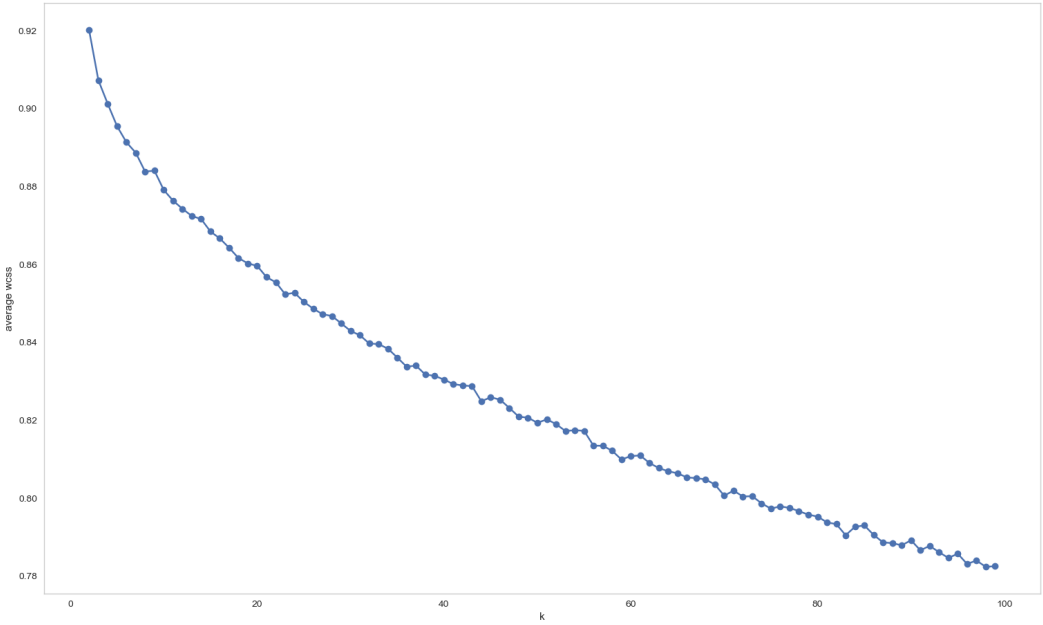


Figure 1. The ‘elbow’ method for determining an optimal number of clusters. The average within-cluster sum of squared errors. Ideally an inflexion point at low error indicates a good number of clusters. Plotted here is a range from 2 to 100 clusters for the ‘KOS’ dataset. No optimal number of clusters could be found

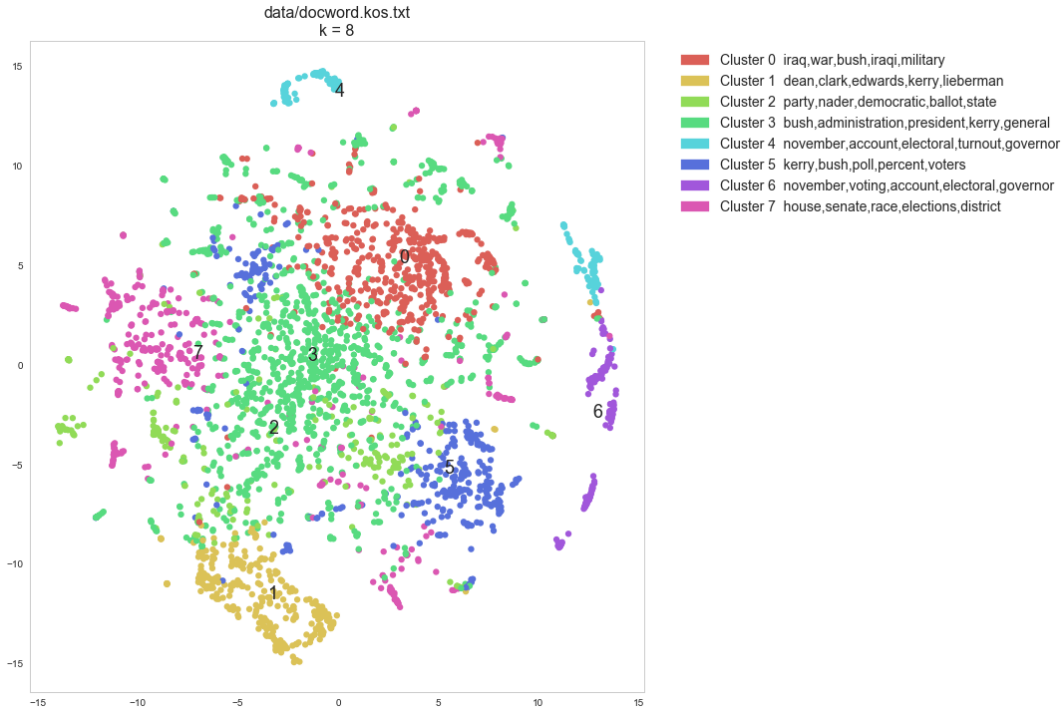
To further validate the found topics on the clusters from the K-Means approach, a complementary term enrichment analysis was also performed. The hypergeometric test provides a probability of finding a particular term within a group (cluster from K-Means) given the frequency of that term in its group and in the whole corpus.

$$\text{Equation 4)} \quad P(x|N,m,k)=\frac{\binom{m}{x}\binom{N-m}{k-x}}{\binom{N}{k}}$$

$$\text{Equation 5)} \quad P(\text{at least } x|N,m,k)=\sum_{i=x}^{min(k,m)} P(i|N,k,m)=1-\sum_{i=0}^{x-1} P(i|N,k,m)$$

The probability of finding the term *x* in the group is given by equation 1, where *m* is the absolute frequency of a particular term in the corpus, *k* is the number of different terms in a defined group and *N* is the total number of terms in the corpus. The probability is then adjusted by summing over the times a term is searched and subtracted from 1 to express it as a positive probability as defined in equation 2.

a)



b)

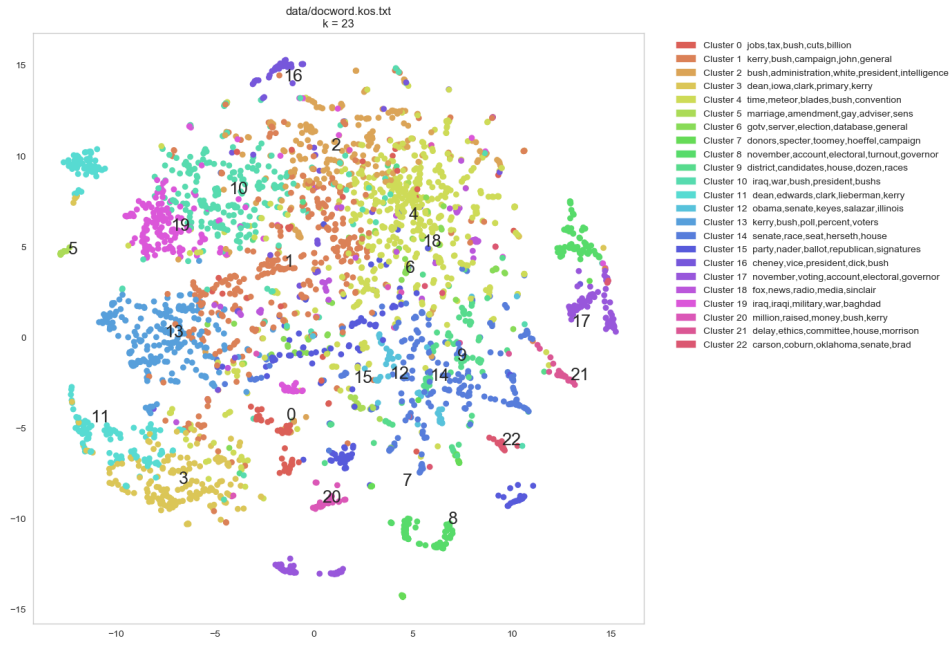


Figure 2. tSNE (t-Distributed Stochastic Neighbor Embedding) representation of 8 K-Means clusters (Figure 2a) and 23 K-Means clusters (Figure 2b) of the *KOS* dataset. tSNE is technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets

Discussion

The application of reduced-ranked SVD, also known as Latent Semantic Analysis (LSA) in the context of text mining is particularly well suited to the analysis of document-term matrices due to their inherent sparseness. The dimensionality reduction will get rid of many columns for which the word count is zero or very low in most documents and therefore contribute very little to the general variability. The number of components that the original matrix should be reduced to is subject of discussion. In this study it was selected such that the transformed matrix preserves at least 75% of the variability.

The LSA reduced matrix makes in addition the subsequent K-Means step less computationally expensive. An exploratory assessment in search of an optimal number of clusters for K-Means can be done by looking at the within-cluster errors for a range of *k* values. However in most cases a clear *k* will not be possible to get. In the context of topic extraction, and for the studied datasets, this might indicate that the clusters do not have a very clear separation, but rather there are relevant terms present in several clusters making topics intermingled between clusters.

Another possible approach could be to implement clustering methods that do not require a predefined *k*, but instead look at regional densities and try to learn an optimal number of clusters. In this respect, the algorithms DBScan and Means Shift were tested without yielding good results. They seemed not able to discern clusters in the population of documents.

High dimensional datasets, NYTIMES and PUBMED pose a challenge in terms of computational cost. In these cases an alternative approach involves using a distributed computing framework such as Apache Spark

References

Karl A, Wisnowski J, Rushing WH. A practical guide to text mining with topic extraction. *Computational Statistics.* 2015, 7:326–340

Baker K. Singular Value Decomposition Tutorial. 2005 March www.ling.ohio-state.edul-kbaker/

Van der Maaten L. Visualizing Data using t-SNE
Journal of Machine Learning Research. 9 (2008) 2579-2605