Blind Source Separation for Text Mining

Abdelghani Ghazdali, Abdelmoutalib Metrane and Amal Ourdou
LIPIM, University Sultan Moulay Slimane, Beni-Mellal
E-mail: a.ghazdali@gmail.com, ab.metrane@gmail.com, amal.ourdou@gmail.com

Abstract. Blind Source Separation (BSS) was originally developed for signal processing applications. It has been proven out that Independent Component Analysis (ICA) which is the technique used for separating independent sources, is a powerful tool for analyzing text document data as well, if the text documents are presented in a suitable numerical form. This opens up new possibilities for automatic analysis of large textual data bases: detecting the topics present in the corpus and grouping the documents accordingly or in other words Clustering documents, hence achieving two tasks of Text Mining at the same time using only one algorithm. In our study we use an appropriate BSS approach along with new weighting distance to transform the textual data to achieve higher level of accuracy.

1. Introduction

Text Mining has always and is still an active research field, due to the huge amount of text produced every single second, whether in physical copies or digitally that need to be analyzed and properly interpreted. Stats claim that almost 80% of the existing text data is unstructured, meaning it’s not organized in a predefined way, it’s not searchable, and it’s almost impossible to manage. In other words, it’s just not useful. Being able to organize, categorize and capture relevant information from raw text data is a major concern and challenge.

To define it we can say that Text mining is the automatic process, technologies and techniques used to extract valuable insights from unstructured text, by transforming data into normalized and structured information that machines can understand then using Machine Learning to classify or clustering text documents by sentiment, topic, and intent, etc. Text mining combines notions of statistics, linguistics, and machine learning to create models that learn from training data and can predict results. The obvious challenge for Text mining compared to data mining is that in data mining the data is assumed to be already stored in a structured format. On the other hand, the problem with modeling text is that it is messy, we cannot really use raw text directly and so the pre-processing operations in text mining systems center on the extraction and the retrieval of meaningful features for natural language documents and it is the most important step in the Text Mining process because if chosen right, these Pre-processing steps can improve the quality of the results drastically.

There are multiple tasks for Text mining, for example, Text Classification which is the process of assigning categories/tags to text documents, it is used for topic analysis, sentiment analysis, language detection and intent detection, etc. A second task is Text Extraction where we extract specific pieces of data from a text, like keywords, entity names, addresses, emails, etc. Also, Text clustering, where we have no training data and we only cluster documents according to unknown topics.
In this paper we introduce a new process for Text Mining where we achieve both Text clustering and Topic detection, using Blind Source Separation, which is an unsupervised Machine Learning technique used mostly within the signal processing community, however it is a great method for detecting hidden trends in data, hence its usefulness in Text mining. BSS aims to separate a set of source signals from a set of mixed signals without or with very little information about the source signals or the mixing process. Independent component analysis (ICA) is one of the first sort of generic algorithm that were able to solve this problem. ICA assume the statistical independence of the sources (hence the name) and give good results to solve the problem. ICA was proven to be a powerful tool to finding latent structure in high dimensional data [1]. The idea is expressing multidimensional observations as a combination of unknown latent variables that are statistically independent of each other. Many attempts were made to use ICA in the text mining field [2, 3, 4], However serious changes needed to be done in the pre-processing steps to improve the quality of the results.

In this paper we will use various pre-processing techniques for cleaning the textual data and use a new scoring system for transforming text data into appropriate input then use a BSS approach [5] to accomplish both document clustering and Topic detection. The present paper will be organized as follow: In ”Approach” we detail each step in this Text mining process, beginning from cleaning the data to the final step. Then in ”Results” we present the Topic detected in the chosen database. and finally a conclusion is given.

2. Approach

The Text Mining Process is not easy and go through multiple steps. The intuition in our process is that documents are similar if they have similar content and that from the content alone we can learn something about the meaning of the document. In the following we present and detail the different steps used in our approach for Text Mining. Figure. 1 summarizes this process.

![Figure 1. Summary of the proposed approach](image)

2.1. Pre-processing step

The first step to modeling text data is cleaning it and managing vocabulary. Each document in the corpus will be modeled as a vector where the features are the weight of each word or token present in the corpus for that document. But as the vocabulary size increases, so does the vector representation of documents. You can imagine that for a very large corpus, such as thousands of books, that the length of the vector might be thousands or millions. Further, each document may contain very few of the known words in the vocabulary. Hence cleaning the Text
and decreasing the number of vocabulary in the corpus as well as maintaining the general idea of the document is a necessity.

There are simple text cleaning techniques that can be used as a first step, such as:

- **Ignoring case**: Lowercasing all text data, although commonly overlooked, is one of the simplest and most effective form of text preprocessing. It decreases the number of vocabulary in each document for example: Canada, canadA, CANADA can all be transformed to canada.

- **Noise removal**: is about removing characters, digits and pieces of text that hold no meaning, for example changing ”Trouble!” or ”..trouble..” to ”trouble”.

- **Stop words**: are a set of commonly used words in a language. Examples of stop words in English are ”a”, ”the”, ”of”, etc. The intuition behind omitting stop words is that, by removing low information words from text, we can focus on the important words instead.

- **Stemming**: works by cutting off the end or the beginning of the word, taking into account a list of common prefixes and suffixes that can be found in an inflected word for example: transforming ”Studying” to ”study” and so on.

All these steps are intended to decrease the number of vocabulary that will be transformed later into appropriate numerical input, as well as leaving only the important ones. For this pre-processing steps we used tool kits such as nltk (natural language tool kit) and the scikit-learn package in python.

### 2.2. Transforming data

The problem with modeling text is that it is messy, and every algorithms prefer well defined fixed-length inputs and outputs. These algorithms cannot work with raw text directly; the text must be converted into numbers. Specifically, vectors of numbers. In language processing, the vectors are derived from textual data, in order to reflect various linguistic properties of the text. This is called feature extraction or feature encoding.

Once a vocabulary has been chosen using the pre-processing methods described above, the frequency of words in documents needs to be scored. In this study we chose to work with the well know TF-IDF scoring system and its modified version [6].

- **TF-IDF**

  Term Frequency-Inverse Document Frequency, or TF-IDF for short is a scoring system that weight words by how frequent the word appear in a document as well how often they appear in all documents, so that the scores for frequent words that are also frequent across all documents are penalized. It is calculated as follows

  \[
  TF_{t,d} = \frac{count_{t,d}}{N} \quad (1)
  \]

  with \( count_{t,d} \) is the number of times term \( t \) appears in document \( d \) and \( N \) is the total number of terms in the document

  - Inverse Document Frequency: is a scoring of how rare the word is across documents.

  \[
  IDF_t = \log \left( \frac{M}{M_t} \right) \quad (2)
  \]

  with \( M \) Total number of documents and \( M_t \) Number of documents with term \( t \) in it.
The score is then calculated by multiplying $TF(t) \times IDF(t)$. The TF-IDF is computed for each term in each document. The scores are a weighting where not all words are equally as important or interesting. The scores have the effect of highlighting words that are distinct (contain useful information) in a given document.

- **MTF-IDF** The modified TF-IDF scoring system that was proposed in [6], consider the proportion of the total count of a term in all documents (denoted by $T_t$) to the total token count of the corpus (denoted by $T_c$). It is calculated as follows:
  
  $$ mTF_{t,d} = \frac{TF_{t,d} \times \log \frac{\sqrt{T_c}}{T_t}}{\log \left( \sum_{i=1}^{n} tf_{t,d}^2 \times \left( \frac{\text{length}_d^2}{\sqrt{T_c}} \right) \right) } \tag{3} $$

  Where

  $$ T_t = \sum_{d=1}^{D} TF_{t,d} \text{ where } TF_{t,d} > 0 \text{ and } T_c = \sum_{d=1}^{D} \sum_{t} TF_{t,d} \tag{4} $$

  - modified Inverse Document Frequency:

  $$ mIDF_t = \log \left[ \frac{N}{1/((N-DF_t) + 1)} \right] \tag{5} $$

  Where $(N - DF_t)$ represents the number of documents where the term $t$ does not appear.

  The score is then calculated by multiplying $mTF_{t,d} \times mIDF_t$.

After calculating the term weight for each token in each document using the different scoring systems we have now a matrix with each row as a document where the features are the words and the values are the scores calculated using TF-IDF or its modified version. Next we will show how we can use this matrix as our input for the BSS technique to achieve Topic Detection and Document clustering.

### 2.3. BSS

Blind source separation is the techniques to separate a bunch of mixed sources of any type without having much information about the original sources nor the mixing environment, hence the word blind. Independent component analysis (ICA) is one of the first sort of generic algorithm that were able to solve this problem.

We observe these signals using $N$ sensors hence we obtain $N$ observation signals $x_i(t)$ with $i = 1...N$, that are the linear mixture of the sources $s(t)$. The sensors must be separated from each other so that each one pick up the sources with different weights. Thus the mathematical model can be written as follows:

$$ x(t) = As(t) \tag{6} $$

where $A$ is the unknown mixing matrix and $x(t)$ and $s(t)$ are respectively the observation vectors and the source vectors. The objective is to recover the original sources $s(t)$, knowing only the observed vector $x(t)$. To do that we have to find a de-mixing matrix $B$ (up to a scale and permutation indeterminacies) that is as close as possible to $A^{-1}$. This enable an estimate $y$, of the independent source to be obtained:

$$ y(t) = Bx(t) \tag{7} $$
In the case of Topic detection and Document clustering we can consider the model 7 as a soft-clustering problem. Figure 2 illustrate more this idea.

![Figure 2. Soft-clustering and Topic detection using BSS](image)

with $X$ the observed matrix that which is the matrix calculated using one of the weighting methods introduced in section Transforming DATA, the mixing matrix $A \in R^{M \times L}$ with $M$ the number of tokens and $L$ the number of topics, tells the degree of activity of a token in each topic. Each row of the The matrix $S$ is a topic and the $i$-th element of a topic vector have the probability of the $i$-th token in that topic. Hence having the matrix $A$ is equivalent to soft-clustering the documents in term of topics, and the matrix $S$ is equivalent to detecting the topics present in the corpus.

2.4. Softmax
The softmax function [7], is a function that takes as input a vector $z$ of $n$ real numbers, and normalizes it into a probability distribution consisting of $n$ probabilities proportional to the exponentials of the input numbers. That is, prior to applying softmax, some vector components could be negative, or greater than one; and might not sum to 1; but after applying softmax, each component will be in the interval $(0,1)$, and the components will add up to 1, so that they can be interpreted as probabilities.

In this study Softmax is often used to map the non-normalized outputs $A$ representing the Documents times Topics matrix and $S$ representing the Topics times Tokens matrix of the BSS algorithm to a probability distribution. The softmax function is given by

$$\sigma(z) := \frac{1}{\sum_{j=1}^{n} \exp(\lambda z_j)} \begin{bmatrix} \exp(\lambda z_1) \\ \vdots \\ \exp(\lambda z_n) \end{bmatrix}, \lambda > 0 \quad (8)$$

where $\lambda$ is referred to as the inverse temperature constant.

3. Results
3.1. Data
The data used in this work are the articles present in the Wikipedia website, each article is considered as one document and the collection of these documents are the corpus.

3.2. Detected Topics
Table. 3 illustrate the 10 words or token that highly represent the Topics detected from The corpus using the modified version of TF-IDF.
• Topic 1 is about Machine Learning
• Topic 2 is about video games and the use of Artificial intelligence in this field
• Topic 3 is about black holes

**Figure 3.** Topics detected using the modified version of TD-IDF

| Topic 1     | Topic 2     | Topic 3     |
|-------------|-------------|-------------|
| Learn       | Game        | Black       |
| Machine     | Video       | Hole        |
| Data        | Intelligence| Science     |
| Knowledge   | Computer    | Mass        |
| Method      | use         | Sun         |
| AI          | Play        | Gravity     |
| Algorithm   | Pattern     | Galaxy      |
| Unsupervised| Character   | Star        |
| Computer    | Movement    | Light       |
| Mining      | Genre       | Space       |

4. Conclusion
We presented in this study a step by step approach to achieve both Document clustering and Topic detection using one algorithm which is BSS. Initially BSS was used by the signal processing community, however due to its usefulness it was extended to a number of other scientific fields. Although it proved its strength dealing with Text document, one disadvantage of this approach is that the number of Topics present in a corpus should be known before hand. Other than that BSS can successfully Detect the Topics present in a corpus as well as regrouping the documents accordingly.

References
[1] Wang F, Li H and Li R 2006 Data mining with independent component analysis 2006 6th World Congress on Intelligent Control and Automation vol 2 (IEEE) pp 6043–6047
[2] Bingham E et al. 2003 Advances in independent component analysis with applications to data mining (Helsinki University of Technology)
[3] Kolenda T, Hansen L K and Larsen J 2002 Computer Science Preprint Archive 2002 95–100
[4] Kolenda T, Hansen L K and Sigurdsson S 2012 Advances in Independent Component Analysis 235
[5] Keziou A, Fenniri H, Ghazdali A and Moreau E 2014 Signal Processing 104 319–324
[6] Sabbah T, Selamat A, Selamat M H, Al-Anzi F S, Viedma E H, Krejcar O and Fujita H 2017 Applied Soft Computing 58 193–206
[7] Gao B and Pavel L 2017 arXiv preprint arXiv:1704.00805