# DIGIT.B4 – Big Data PoC

## DG GROW

## D03.03.Text Mining Models

everis Spain S.L.U

# Table of contents

## List of figures

# 1 INTRODUCTION

## 1.1 Context of the project

The objective of the proof of concept showcasing the use of big data in the procurement domain, in cooperation with DG GROW, is to prove the usefulness and policy benefits that big data can bring.

This proof of concept shall also demonstrate the use of natural language analysis techniques to check the compliance of the transpositions sent by European Member States related to EU directives. In the context of the PoC, one directive and its respective national transpositions will be analysed, with the objective of supporting the manual checks currently done by European Commission staff.

## 1.2 Objective

The purpose of this document is to describe the processes carried out during the modelling phase of the CRISP-DM methodology. In this phase, segmentation and classification algorithms are used in order to establish a relationship between the articles of a legal directive and the paragraphs of its transposition.

# 2 MODELLING

## 2.1 Introduction

The goal of the modelling is to assign the percentage similarity between the different articles of the directive and the paragraphs of the transposition in each one of the languages.

The first step of the modelling consists of identifying the best-fitting algorithm for the purpose of the analysis. Therefore, different algorithms have been tested to find the best model:

- Latent semantic algorithm for discovering hidden concepts in document data;
- Classification algorithm, a technique from machine learning that scales well to high-dimensional data;
- Topics discovery algorithm, specifically algorithms called "correlated topic models" (selected for this PoC).

## 2.2 Latent semantic

### 2.2.1 Algorithm

Latent semantic indexing (LSI) is a method for discovering hidden concepts in document data. This algorithm uses a technique that maps out queries and documents into a space with “latent” semantic dimensions. Thus, the goal is to extract the conceptual content from a body of text by establishing similarities or a semantic relationship between terms.

An abstract of the LSI algorithm may be:

1. **Cleaning the TDM**, which aims to avoid *noise*, *sparsity*, *huge size,* etc. The original algorithm uses a weighted low-rank approximation to the TDM, trying to minimise its Frobenius norm.

2. **Rank-reduced,** using a least-squares method called **singular value decomposition** (SVD). The projection into the latent semantic space is chosen so that the representations in the original space are changed as little as possible when measured by the sum of the squares of the differences.

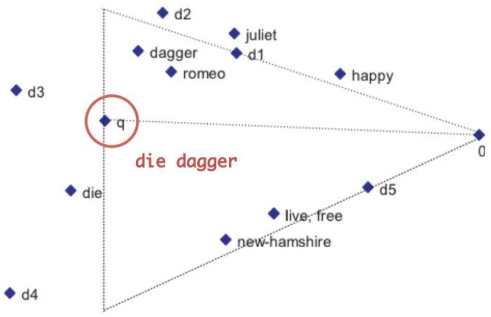


**Figure 1 - LSI algorithm**

### 2.2.2 Application

Every vector of keywords (grouped by article) is executed with every paragraph of the transposition to get each distance and then rotate this distance to create a percentage of compliance.

## 2.3 Classification

### 2.3.1 Algorithm

The main objective of classification algorithms is to identify to which one of a set of categories a new observation belongs to. Specifically, the Winnow algorithm (decision tree) has been used to split the data set into branch-like segments.

Every decision tree has three parts:

1. **Decision nodes:** Indicate that a decision has to be made in this moment of the process. They are represented by squares.
2. **Chance nodes:** Indicate a random event appearing in this moment of the process. They are represented by circles.
3. **Branches:** Show the different paths you can follow when you make a decision or a random event occurs.

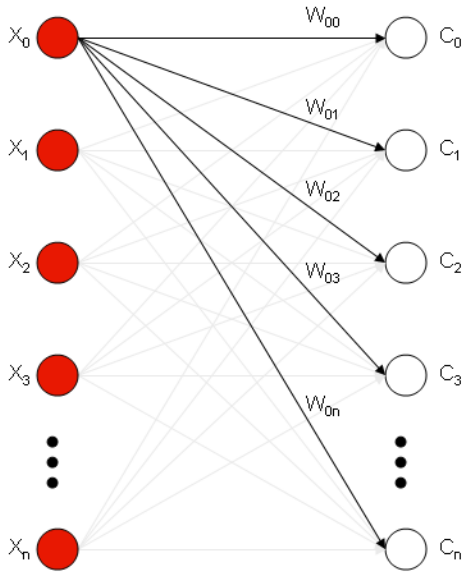


**Figure 2 - Winnow algorithm**

The decision rule to form the branches is based on a method that extracts the relationship between the target field and one or more variables.

The **steps** to make a decision tree analysis follow below:

1. Define the problem;
2. Draw the decision tree;
3. Assign probabilities to random events;
4. Estimate the results for each possible combination of alternatives;

5. Solve the problem obtaining the best split.

### 2.3.2 Application

Considering that each article of the directive is a different target, the algorithm will build rules to classify multiple targets.

These rules are applied to the paragraphs of the transposition to obtain the compliance with every article.

## 2.4 Topics discovery

### 2.4.1 Algorithm

The original developers of LDA wrote:

*"In technical terms, a topic model is a generative probabilistic model that uses a small number of distributions over a vocabulary to describe a document collection. When fit from data, these distributions often correspond to intuitive notions of topicality".*

As in the previous model, the input is a TDM or a DTM (Term/Document or Document/Term matrix) so it assumed that the words in a document are exchangeable and their order is not important for the document summary (a *bag-of-words algorithm*).

A summary of iterative algorithms could be made with the following steps:

1. The initial **number of topics** expected is fixed. In many mathematical classification models it is necessary to previously set the topics to detect. The fixed number could be pondered or taken from a previous analysis. This number would be called $N$.

2. **Every word** is **assigned** to a temporary topic according to some function. This function will be the Dirichlet distribution (a multivariate generalisation of *Beta distribution*). This assignment proceeds as follows:

    a. For each topic and each document, both distributions are used: $Dir(\vec{\alpha})$ and $Dir(\vec{\beta})$, they would be called $T$ and $D$.

    b. All the topics previously generated are distributed, creating geometrical figures with a centroid. These 'centroids' are moved according to a multinomial distribution, whose parameters are $T$ and $D$. They would be called $M_T$ and $M_D$.

3. Each topic is properly **labelled,** because initially the algorithm gets numbered topics.

4. The previous steps are repeated for multiple iterations. In each iteration, the centroids move towards a convergence point. The iterations stop when the centroids do not move anymore compared to the previous iteration (the algorithm "converges"). The following graph shows the steps above by summing up the **assignment algorithm** with the previous notation:

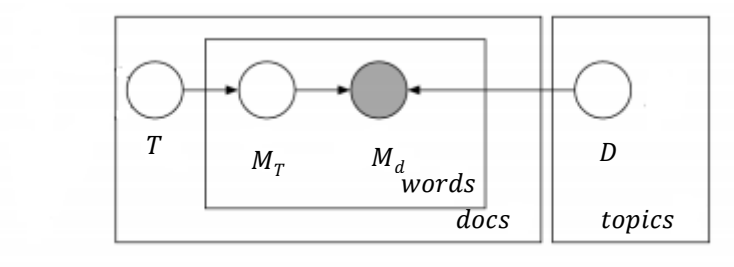


**Figure 3 - LDA algorithm**

In order to calculate the topics probability distribution in words and documents, **some information about several parameters** must be estimated.

Due to the computational complexity of their maximum likelihood estimation, a **Gibbs sampling** is used (a Markov chain Monte Carlo algorithm) for obtaining them within the LDA.

### 2.4.2 Application

Segments are made, forcing the algorithm to assign a topic for each article of the directive. Several iterations are necessary to build the oriented LDA model because the original keywords didn't have a high enough frequency to represent to the articles. The solution to this problem is to weigh the keywords, increasing their relevance artificially.

Once the segments are defined, each paragraph is exploited by the model to get the distance to each segment (representing each article).

# 3 RESULTS

Finally, the LDA model is selected and built with ten articles from the different languages.

The following figure shows the results obtained through the LDA model with the keywords weighted. Each blue circle (or cluster) represents the abstracts that are classified under a topic and the terms with the highest frequency inside them.

For example, topic number 5 is article 6 because the principal keyword is ‘art_6’.

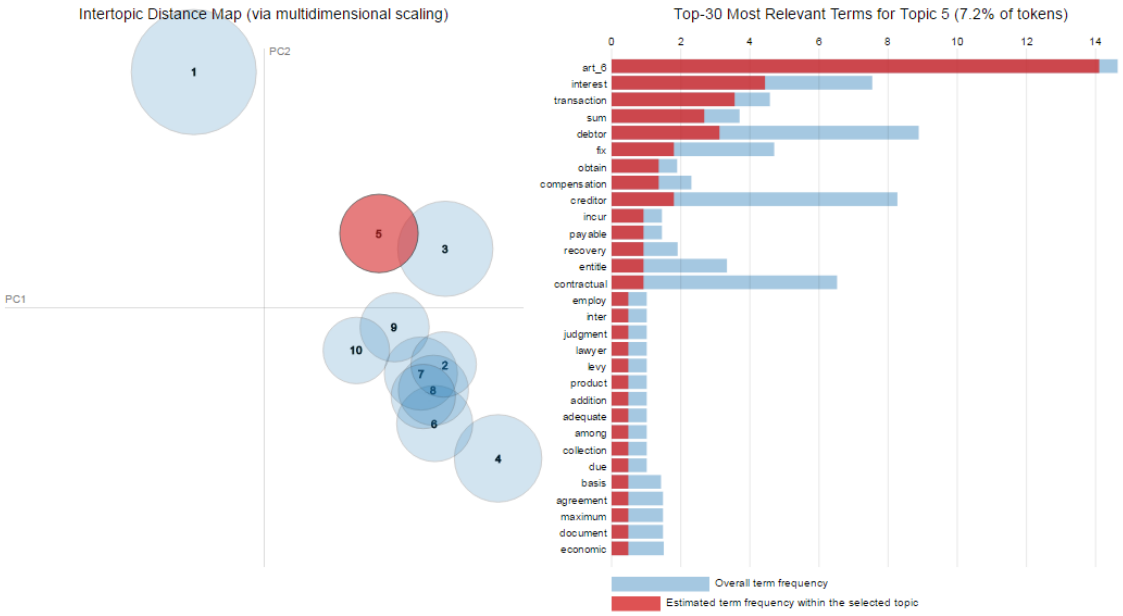


**Figure 4 - Examples of topic assignment**

After the data cleaning process and the topic modelling, it is possible to assign the probabilities of belonging or compliance of each paragraph to the articles. The figure below shows the matrix of probabilities from 15 paragraphs:

| | Art1 | Art2 | Art3 | Art4 | Art5 | Art6 | Art7 | Art8 | Art9 | Art10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Paragraph1 | 14% | 8% | 8% | 10% | 12% | 8% | 8% | 10% | 12% | 10% |
| Paragraph2 | 10% | 8% | 10% | 10% | 8% | 8% | 8% | 10% | 10% | 18% |
| Paragraph3 | 8% | 13% | 7% | 24% | 13% | 4% | 8% | 3% | 6% | 16% |
| Paragraph4 | 10% | 8% | 9% | 8% | 13% | 10% | 8% | 13% | 9% | 12% |
| Paragraph5 | 10% | 10% | 10% | 10% | 10% | 12% | 10% | 10% | 10% | 10% |
| Paragraph6 | 10% | 10% | 10% | 10% | 8% | 15% | 10% | 7% | 8% | 13% |
| Paragraph7 | 12% | 9% | 9% | 9% | 9% | 12% | 9% | 12% | 9% | 9% |
| Paragraph8 | 31% | 9% | 9% | 6% | 4% | 6% | 5% | 9% | 7% | 11% |
| Paragraph9 | 10% | 10% | 10% | 10% | 10% | 10% | 10% | 10% | 10% | 10% |
| Paragraph10 | 10% | 10% | 10% | 8% | 8% | 10% | 8% | 12% | 8% | 12% |
| Paragraph11 | 20% | 5% | 8% | 6% | 9% | 24% | 8% | 7% | 7% | 6% |
| Paragraph12 | 13% | 9% | 9% | 11% | 9% | 11% | 13% | 9% | 9% | 11% |
| Paragraph13 | 14% | 9% | 9% | 9% | 12% | 9% | 9% | 9% | 9% | 9% |
| Paragraph14 | 9% | 9% | 9% | 9% | 9% | 9% | 12% | 9% | 12% | 12% |
| Paragraph15 | 10% | 10% | 10% | 10% | 10% | 10% | 10% | 12% | 10% | 10% |

**Figure 5 - Example of compliance matrix**

The result of the model is a visual application of the three paragraphs with the highest probability in each article.


**Figure 6 - Example of the application**

# 4 CONCLUSIONS AND NEXT STEPS

Due to the complexity of the objectives of the PoC and the non-ideal scenario for applying text-mining techniques, as highlighted in the conclusions of the data linguistic understanding document (D03.01), further work is required to refine the algorithms already tested or, if necessary, to try additional techniques that can lead to more accurate results. This further work will be carried out in the next phases of this project.