University of Turin

Department of Computer Science

Natural Language Technologies Project

Implementation of a topic modelling system

Authors:
Christian Quaranta
Giulia Coucourde
Leonardo Magliolo

Academic Year 2022/2023

Introduction

The goal of the project is to identify in an unsupervised way the major themes found in large collections of documents (Topic Modelling).
The project consists implementing a Latent Dirichlet Allocation (LDA) model using Gensim library for natural language processing.
LDA was chosen as the model to address the topic modelling task because of its effectiveness, scalability, interpretability and applicability in unsupervised settings.
Gensim was chosen for its versatility in NLP and ease of use; it allowed us to experiment with different configurations and parameters to improve performance.
The paper will explain the methodological approach, the implementation of the LDA model through Gensim and the results obtained on test datasets, demonstrating the effectiveness of the system in discovering themes consistently and quickly.

Design

Pre-processing:

It was chosen to implement at the pre-processing stage the removal of stopwords from the corpus used training the LDA model, and then words not classified by the POS-tagger as nouns were removed; this was done to increase the accuracy of the model and make the computed themes more interpretable.

Latent Dirichlet Allocation:

LDA is a generative statistical model used to discover hidden topics within large collections of texts. The LDA generative process assumes that each document is a mixture of various topics and each topic is a probability distribution over vocabulary terms. goal of LDA is then to estimate these probability distributions from the observed data (the documents).

The data generation process in the LDA model can be summarized as: For each document:

- Generates a probability distribution on the topics.
- For every word in the document:
  - Generate a topic from this distribution.
  - Generate a word from this probability distribution of terms for the selected topic.

goal LDA training is to estimate the parameters that maximize the probability of observing the data, that is, the documents actually in the corpus.

Training is generally done using the Gibbs inference algorithm or other optimization techniques. During this process, the LDA model attempts to optimally assign arguments to terms and documents so as to maximize the likelihood of the observed data.

Once trained, the LDA model can be used to assign topics to new documents, identify the most relevant terms for each topic, and generally provide an interpretable representation of the thematic structures present in the text corpus.

Parameters and hyper-parameters of the LDA model are listed below: LDA parameters:

- Probability distribution of topics per document: This parameter represents the mixture of topics within each document. It is a probability distribution over the K topics, where K is the number of topics specified as the hyper-parameter.
- Probability distribution of terms by topic: This parameter represents the probability distribution of terms within each topic. It indicates which terms are most representative for a specific topic.

LDA hyper-parameters:

- Number of Topics (K): This is the main hyper-parameter of LDA and specifies the number of topics we want to discover in the corpus. The value of K must be chosen a priori, and it is important to find an appropriate value to obtain meaningful results.
- Alpha (α): The hyper-parameter α controls the probability distribution of topics per document. A higher value of α will make documents more focused on a number limited number of topics, while a lower value of α will make the documents more evenly distributed among all topics. Typically, a low value of α is used to have a more even distribution of topics for documents.
- Beta (β): The hyper-parameter β controls the probability distribution of terms for topic. A higher value of β will make the arguments more focused on a number limited number of terms, while a lower value of β will make the arguments more evenly distributed among all the terms. Typically, a low value of β is used to have a more even distribution of terms for the arguments.
- Iterations: This hyper-parameter specifies the number of iterations or epochs during the training process. A larger value of iterations can improve model convergence, but it also increases the training time.

Implementation

In addition to the native Python libraries, the following were used:

- Numpy and Pandas: To represent datasets in matrix form.
  The use of Numpy and Pandas reduced the computation time of the solutions and standardized the implementation to one of the most common programming standards for AI systems in Python.
- NLTK: Used in preprocessing to perform tokenization, stopword removal and POS-tagging operations.
- Gensim: Used to generate LDA models from a corpus and to assess goodness from models by perplexity and coherence index.
- Matplotlib: Used to automatically produce graphs useful for debugging and To data analysis.
- PyLDAvis: Used to produce an Intertopic Distance Map via multidimensional scaling of LDA models computed by means of Gensim

Dataset

The reference dataset used is New York Times Annotated Corpus [1] containing over 1.8 million articles written and published by The New York Times between January 1, 1987 and June 19, 2007 with article metadata provided by The New York Times Newsroom, The New York Times Indexing Service, and nytimes.com personal online production.

The corpus includes:

- Over 1.8 million articles (excluding news agency articles that appeared during the reporting period).
- Over 650,000 abstracts of articles written by library scientists.
- Over 1,500,000 items manually tagged by library scientists with tags drawn from a normalized indexing vocabulary of people, organizations, places, and subject descriptors.
- Over 275,000 algorithmically coded items that have been manually verified by the online production staff at nytimes.com.

Given the size of the dataset, we opted to use a small portion of it. We implemented functions that extracted 'abstracts' by balancing them according to different categories of 'Section_name' (referred to within the implementation as 'label' or 'category'). This approach allowed the analysis of texts belonging to macro-topics in a balanced manner.

Performance testing

Metrics used:

- Perplexity: measures how well the LDA model is able to predict documents in the training corpus.
  It represents a measure of the model's "confusion" in assigning topics to documents.
  A lower perplexity value indicates a better predictive ability of the model; the goal is to reduce it during LDA training.
- Coherence index: measures the semantic consistency of the arguments discovered by the model.
  A higher coherence score indicates better semantic coherence of arguments; the goal is to maximize the score for more meaningful and interpretable arguments.
- Trade-off formula: As observed experimentally from the data computed from the models calculated from the implementation, increasing the number of topics assigned to LDA decreases perplexity at the expense of decreasing coherence index.
  In order to identify the model having 'number of topics' that minimized perplexity and maximized coherence index simultaneously, it was useful to consider the following procedure:
  - The two metrics are normalized such that the minimum value in modulus is associated with 0 and the maximum in modulus with 1, by means of the following function:

    ```
    def   normalize_value_array(array):
       min_val  =  np.min(np.abs(array))
       max_val = np.max(np.abs(array))
       return (np.abs(array) - min_val) / (max_val - min_val)
    ```
  - Finally, the two normalized metrics are multiplied to obtain a relative index, between 0 and 1, which determines how well the model maximizes the modulus of both metrics relative to other models having different 'number of topics'.

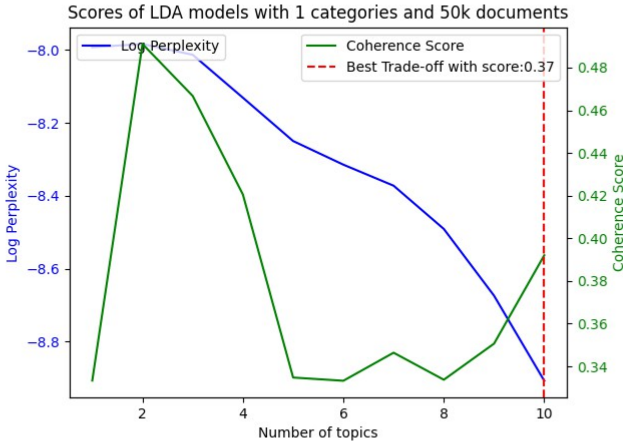
Given the short computation time for a small amount of abstracts (~4 minutes in the worst case for 50,000 abstracts) on machine having the following hardware specifications:

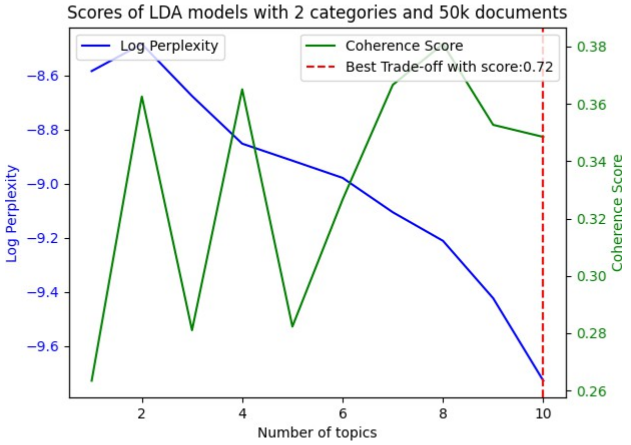
- CPU: AMD Ryzen 7 5800X
- RAM: 32 GBytes, DDR4, 3600 MHz

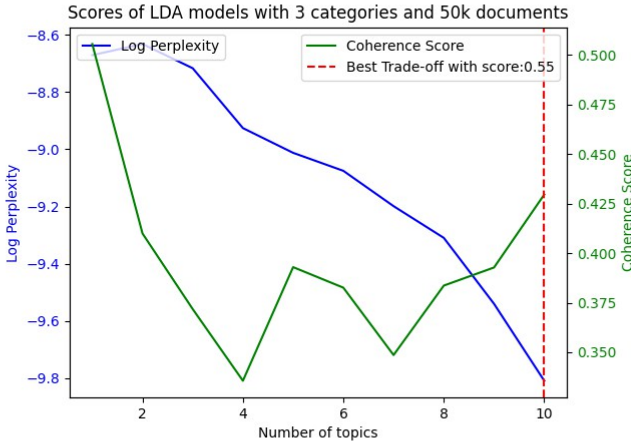
It was possible to compute perplexity, coherence index and trade-off score for models generated from datasets of 50,000 abstracts by varying the number of labels between 1 and 10 and by varying the hyper-parameter 'number of topics' between 1 and 10.
All 100 generated models have the same LDA parameters as Gensim:

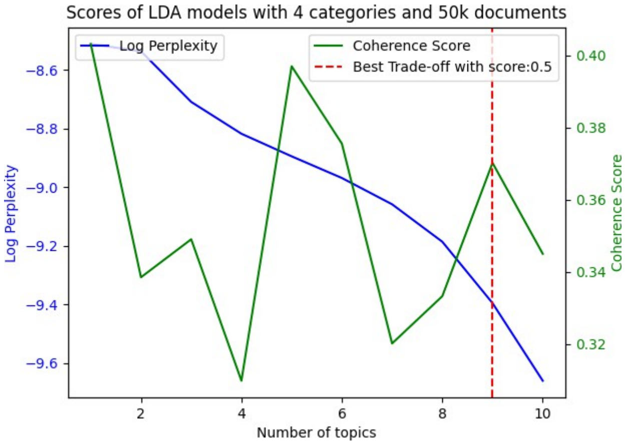
- Random_state: 100
- Update_every: 1
- Chunksize: 10
- Passes: 10 (it was observed that increasing step numbers was not able to significantly increase model performance)
- Alpha: "Auto"

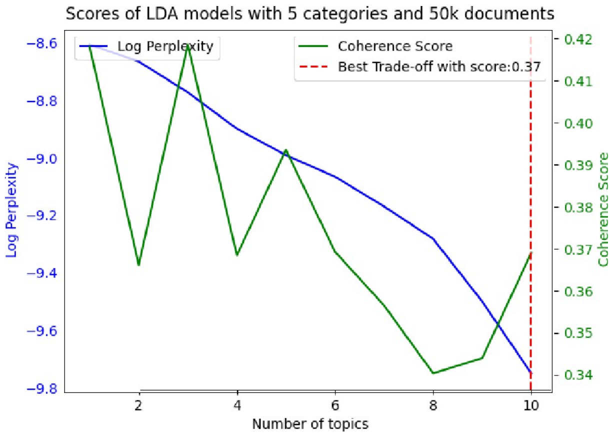
The obtained graphs showing the perplexity and coherence score values in relation to the number of labels and topics are shown below:

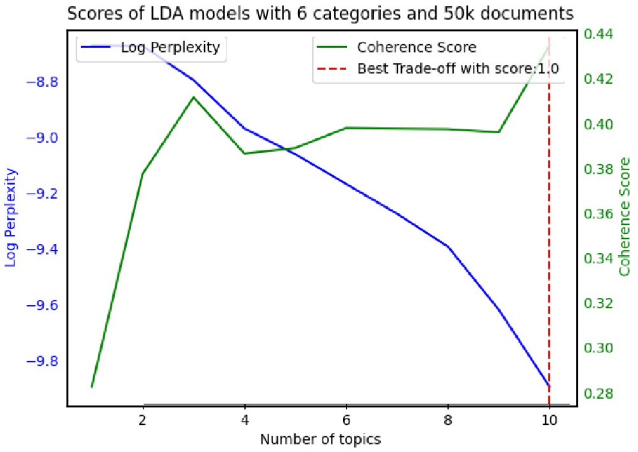


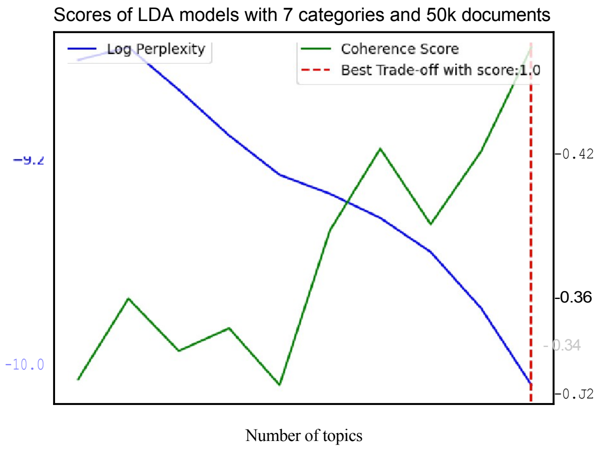


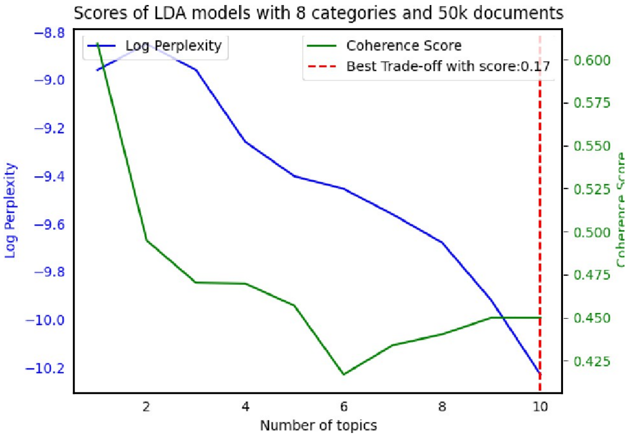


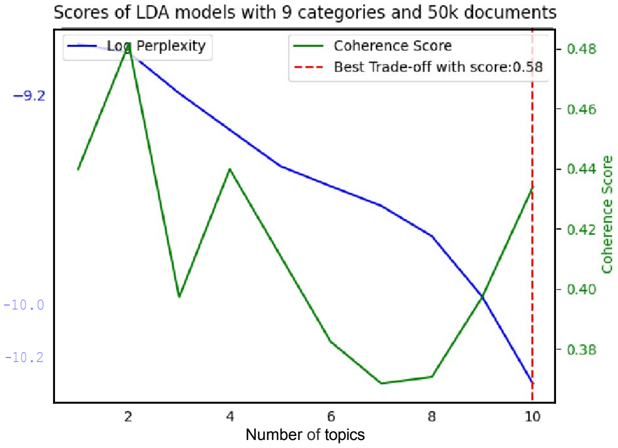


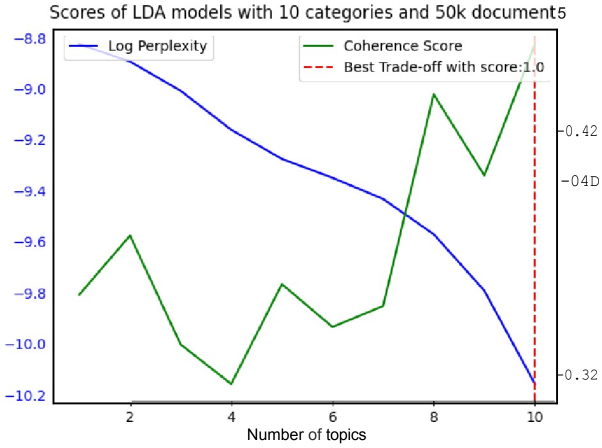

Scores of LDA models with 5 categories and 50k documents
Log Perplexity
Coherence Score
Best Trade-off with score:0.37
−8.6
−8.8
−9.0
−9.2
−9.4
−9.6
−9.8
0.42
0.41
0.40
0.39
0.38
0.37
0.36
0.35
0.34
2
4
6
8
10
Number of topics


Scores of LDA models with 6 categories and 50k documents
Log Perplexity
Coherence Score
Best Trade-off with score:1.0
−8.8
−9.0
−9.2
−9.4
−9.6
−9.8
0.44
0.42
0.40
0.38
0.36
0.34
0.32
0.30
0.28
2
4
6
8
10
Number of topics

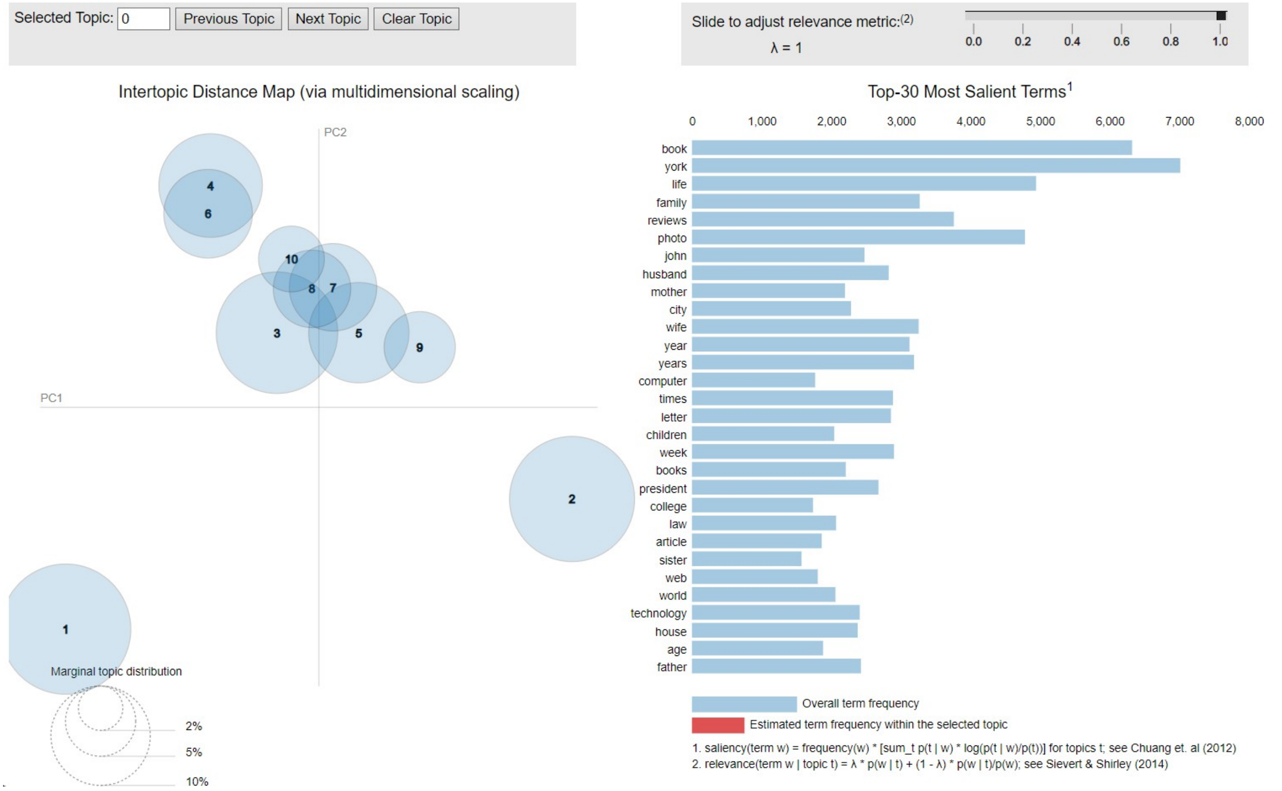

Scores of LDA models with 7 categories and 50k documents
Log Perplexity
Coherence Score
Best Trade-off with score:1.0
−9.2
-10.0
0.42
0.36
0.34
0.32
Number of topics


Scores of LDA models with 8 categories and 50k documents
Log Perplexity
Coherence Score
Best Trade-off with score:0.17
−8.8
−9.0
−9.2
−9.4
−9.6
−9.8
−10.0
−10.2
0.600
0.575
0.550
0.525
0.500
0.475
0.450
0.425
2
4
6
8
10
Number of topics


Scores of LDA models with 9 categories and 50k documents
Log Perplexity
Coherence Score
Best Trade-off with score:0.58
−9.2
-10.0
-10.2
0.48
0.46
0.44
0.42
0.40
0.38
2
4
6
8
10
Number of topics


Scores of LDA models with 10 categories and 50k document5
Log Perplexity
Coherence Score
Best Trade-off with score:1.0
−8.8
−9.0
−9.2
−9.4
−9.6
−9.8
−10.0
−10.2
0.42
04D
0.32
2
4
6
8
10
Number of topics

The Intertopic Distance Map generated by means of PyLDAvis is given below for the model having produced the most convincing results (10 categories and 10 topics):



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Finally, the textual representation of the categories obtained from the above model is shown:

Categories: ['Business Day', 'New York', 'U.S.', 'Opinion', 'World', 'Sports', 'Arts', 'Archives', 'Books', 'Technology']

Pattern: [(0, '0.030*"world"+ 0.020*"internet"+ 0.017*"michael"+ 0.017*"people"+ 0.016*"work"+ 0.012*"lives" + 0.011*"fiction" + 0.010*"director" + 0.010*"mark" + 0.009*"random"'),
(1, '0.124*"book"+ 0.074*"reviews"+ 0.040*"children"+ 0.037*"article"+ 0.028*"author"+ 0.025*"board" + 0.024*"man" + 0.023*"daughter" + 0.023*"member" + 0.022*"story"'),
(2, '0.032*"wife"+ 0.029*"letter"+ 0.026*"president"+ 0.023*"home"+ 0.022*"service"+ 0.021*"services" + 0.018*"st" + 0.015*"james" + 0.014*"church" + 0.014*"ny"'),
(3, '0.098*"family"+ 0.066*"mother"+ 0.047*"sister"+ 0.043*"david"+ 0.038*"grandmother"+ 0.036*"june" + 0.031*"version" + 0.027*"memory" + 0.022*"april" + 0.019*"education"'),
(4, '0.034*"books"+ 0.032*"law"+ 0.029*"age"+ 0.025*"ages"+ 0.024*"woman"+ 0.021*"novel"+ 0.019*"friends" + 0.016*"william"+ 0.012*"joseph"+ 0.012*"rankings"'),
(5, '0.073*"york"+ 0.033*"year"+ 0.030*"times"+ 0.025*"technology"+ 0.025*"house"+ 0.022*"brother" + 0.021*"university" + 0.019*"street" + 0.015*"march" + 0.014*"day"'),
(6, '0.058*"city"+ 0.044*"college"+ 0.028*"power"+ 0.026*"system"+ 0.025*"health"+ 0.013*"records" + 0.013*"access" + 0.013*"sea" + 0.011*"rock" + 0.010*"artists"'),
(7, '0.043*"photo"+ 0.029*"years"+ 0.026*"week"+ 0.022*"father"+ 0.019*"school" + 0.015*"war"+ 0.015*"time" + 0.014*"friend"+ 0.014*"death"+ 0.014*"photos"'),
(8, '0.088*"john"+ 0.064*"computer"+ 0.044*"richard"+ 0.040*"robert"+ 0.034*"boy"+ 0.030*"sales" + 0.028*"software" + 0.025*"stephen" + 0.023*"game" + 0.020*"editor"'),
(9, '0.100*"life"+ 0.056*"husband"+ 0.037*"web"+ 0.031*"weeks"+ 0.030*"july+0.025*"site"+ 0.021*"th" + 0.021*"sites" + 0.017*"mass" + 0.016*"computers"')]

Analysis of results

It was possible to find that the values of perplexity and coherence index varied as the number of topics assigned to the LDA model varied: in accordance with what has been reported in the literature concerning topic modeling, an inversely proportional relationship is observed between the number topics and the perplexity calculated on the model.

Although tests were done with multiple datasets having different numbers of categories was no clear finding about the growth of coherence index near the associated number of categories, this could be explained considering that

- Although the categories listed by the NYTimes are seen as distinct, they may share common themes (as in the case of 'Arts' and 'Books'), or some specific categories may be included in some more general ones (as in the case 'New York' and 'U.S.').
- Additional latent topics may be present in the data, in addition those reported by the abstract categories.

Sources:

1. Evan Sandhaus. The New York Times Annotated Corpus: https://catalog.ldc.upenn.edu/LDC2008T19.