Topic Modeling, Discover Topics and Trends in Computer Science

Abstract

This report examines ”Topic Modelling” using both Latent Dirichlet Allocation (LDA) and Combined Topic Models (CTM). We used data from the dlp, which is a large database containing metadata on computer science (and related) publications. The evaluation of our LDA model led to conflicting results as some outputs made sense while others did not (due to logic and coherence). We determined that before 1990 the topics are mainly basic topics on Computer Science, like Networks or Software. From 1990 to 2009 Computer Vision became more popular. From 2010 onward the topics are mostly about Machine and deep learning, while Computer Vision has still stayed relevant. In direct comparison, the CTM generated slightly more coherent topics than LDA, however the improvement was not significant.

Keywords

Topic Modeling — Latent Dirichlet Allocation — Combined Topic Models

Contents

1. Introduction

This is our report for the last assignment of the Machine Learning for Natural Language Processing 1 course. This exercise is about topic modelling, more specifically about Latent Dirichlet Allocation (LDA) and topic modelling based on pretrained language models (PLM). The main goals of this exercise are to:

- understand how topic modeling is used as a text-mining tool,
- and be able to apply LDA and PLM-based topic models.

2. Data & Task Description

This section discusses the data we worked with and the task we completed. First the data is discussed and then the task is elaborated.

2.1 Data

The task description states that for this exercise, you will work with the dblp: a large database containing metadata on computer science (and related) publications. You will perform topic modelling on the titles (not on the text itself!) of computer science publications to detect important topics and trends in the field. For the exercise, you are given this notebook. It already contains code for downloading the dataset, preprocessing, and for constructing a first topic model. The data can be downloaded manually from the link below. Further the notebook and dblp are linked.

- Link to Data: https://dblp.uni-trier.de/xml/dblp.xml.gz
- Link to Notebook: https://colab.research.google.com/drive/1C8aMJz1zTuX_1a5Np2tgEfrqsD4fqQCz?usp=sharing#scrollTo=TUVy4jyGlVH3
- Link to dblp: https://dblp.org/

The URLs above provide all important data and background information for this exercise.

2.2 Part 1 - Topic Modelling using LDA

The given notebook limits the number of titles to a reasonable amount and divides publications into three time-periods: before 1990, from 1990 to 2009, and 2010 onwards. An LDA-based topic model for the time period “before 1990” is already implemented. Extend the notebook to perform topic modelling on the other two time periods. We encourage you to experiment with different numbers of topics and with different ways of preprocessing. You can increase the number of topics generated by the topic model, but you should not go below 5.

- For each time-period assign a name to each generated topic based on the topic’s top words. List all topic names in your report. If a topic is incoherent to the degree that no common theme is detectable, you can just mark it as incoherent (in other words: no need to name a topic that does not exist).
- Do the topics make sense to you? Are they coherent? Do you observe trends? Discuss in 4-6 sentences.

2.3 Part 2 - Topic Modelling using Combined Topic Models (CTMs)

Bianchi et al. 2021 propose a topic modelling method that makes use of pre-trained language models such as BERT. The authors provide a simple colab tutorial showcasing how to use the CTM library that implements their method. Again, perform topic modelling for the 3 time-periods. This time using the CTMs. Use the same number of topics as before. You can copy and adjust code from the author’s tutorial.

- Again: Assign a name to each topic based on the topic’s top words (for each time-period). List all topic names in your report.
- Bianchi et al. 2021 claim that their approach produces more coherent topics than previous methods. Let’s test this claim by comparing the coherence of the topics produced by CTM with the topics produced by LDA. Describe your observations in 2-4 sentences
- Do the two models generate similar topics? Can you discover the same temporal trends (if there are any)? Discuss in 4-6 sentences.

3. Data Preprocessing

In this section, we discuss the different steps taken to preprocess the data used in this exercise. The task makes use of both LDA and CTM. We will first discuss the steps we took in the first part of the exercise (LDA) and then move on to part two (CTM).

3.1 Latent Dirichlet Allocation

We noticed that all the titles were well formatted and therefor concluded that there was no need to do many pre-processing steps. The only preprocessing steps we took in the exercise were **converting all characters to lowercase** and **removing all punctuation**. To transfer the words to a matrix of token counts (in order for the LDA model to be able to make use of them), we used the class *sklearn.feature_extraction.text.CountVectorizer*.

3.2 Combined Topic Models

The imported module *contextualized_topic_models* is a *WhiteSpacePreprocessing* class. Therefor there was no need to preprocess it. Infact we just needed to import and use it. The stopwords type was set to English because most of the metadata is in English. The next step was to generate the training dataset for the CTM model. We used the *TopicModelDataPreparation* object, which can be used in order to complete this task. We use the contextualized model ”all-mpnet-base-v2” which maps sentences and paragraphs to a 768- dimensional dense vector space and can be used for tasks like clustering or semantic search.

4. Models

In this section, we discuss the different models used in this exercise. The task makes use of both LDA and CTM. We will first discuss LDA and then explain the CTM. Topic modeling is a method for unsupervised classification of documents, similar to clustering on numeric data, which finds some natural groups of items (topics) even when we are not sure what we are looking for. A document can be a part of multiple topics, kind of like in fuzzy clustering (soft clustering) in which each data point belongs to more than one cluster. Topic modeling provides methods for automatically organizing, understanding, searching, and summarizing large electronic archives. Therefore, by annotating the document, based on the topics predicted by the modeling method, we are able to optimize our search process.

4.1 Latent Dirichlet Allocation

In natural language processing, Latent Dirichlet Allocation (LDA) is one of the most popular topic modeling methods. However the applications of LDA need not be restricted to Natural Language Processing. It is a generative statistical model that explains a set of observations through unobserved groups. More specifically, each document is made up of various words, and each topic also has various words belonging to it. The aim of LDA is to find topics a document belongs to, based on the words in it. Observations (words) are collected from documents, and each word’s presence is attributable to one of the document’s topics. Each document will contain a small number of topics. There are 2 parts in LDA:1) The words that belong to a document, that we already know and 2) the words that belong to a topic or the probability of words belonging into a topic, that we need to calculate.

4.2 Combined Topic Models

Standard topic modeling methods deal with two main problems: namely 1) Once trained, most topic models cannot deal with unseen words, this is because they are based on Bag of Words (BoW) representations, which cannot account for missing terms and 2) it is difficult to apply topic models to multilingual corpora without combining the vocabulary of multiple languages (Minmo et al, 2009; Jagarlamudi et al, 2010), making the task computationally expensive and without any support for zero-shot learning. The solution to these issues are Contextualized Topic Models, which are a family of topic models that combine the expressive power of BERT embeddings with the unsupervised capabilities of topic models to get topics out of documents. A pre-trained representation of the documents is passed to the neural architecture and then used to reconstruct the original BoW of the document. Once the model is trained, it can generate the representations of the test documents, thus predicting their topic distributions even if the documents contain unseen words during training.

5. Output

In order to compare the different performances, we decided to use the model to generate ten topics for each period, while training each for ten iterations/epochs.

5.1 Latent Dirichlet Allocation

To implement the model we use the class *sklearn.decomposition-.LatentDirichletAllocation*. The output generated, including the names assigned, is listed below.

- **Before 1990**

  1. problem note optimal functions method technical linear decision solution problems algorithm using **Algorithm**
  2. control new implementation digital optimal linear approach design theory using systems problems **Optimization**
  3. software processing applications finite research parallel digital computer data design theory information **Software**
  4. analysis application languages performance algorithms theory data decision computer networks linear design **Computer Networks**
  5. programming simulation linear problems digital language computer languages approach parallel using design **Programming Languages**
  6. design algorithm data information networks approach performance using parallel digital computer linear **Computer Networks**
  7. computer using theory linear problems algorithms models parallel digital performance decision design **Algorithm**
  8. language recognition sets time pattern solution linear problems using parallel approach problem **incoherent**
  9. logic distributed programs parallel networks using computer functions approach design algorithms theory **Computer Networks**
  10. systems model network linear decision performance computer digital design information theory models **incoherent**

- **From 1990 to 2009**

  1. algorithm new linear problem algorithms optimal robust equations efficient detection optimization multiple **Algorithm**
  2. networks approach nonlinear network models problems neural wireless mobile evaluation scheduling robust **Neural Networks**
  3. systems based distributed nonlinear linear robust control approach optimal adaptive evaluation detection **incoherent**
  4. control analysis methods software development computing robust nonlinear optimal linear adaptive problems **Optimization**
  5. applications scheme web power efficient wireless mobile new robust control networks evaluation **Wireless Networks**
  6. model performance image time graphs parallel digital evaluation algorithms robust scheduling optimal **Computer Vision**
  7. using method dynamic simulation equations nonlinear detection models multiple problems efficient new **incoherent**
  8. adaptive application estimation learning modeling fuzzy theory recognition robust nonlinear control approach **incoherent**
  9. design information management evaluation approach robust development systems network new optimal mobile **Mobile Networks**
  10. data study programming models approach linear multiple evaluation analysis using problems algorithms **Algorithms**

- **From 2010 onward**

  1. networks detection neural linear mobile novel fuzzy recognition computing images deep cloud **Computer Vision**
  2. information framework time problem management scheduling energy dynamic optimal cloud hybrid algorithms **incoherent**
  3. using method deep models optimal energy social machine algorithms learning hybrid feature **Machine and Deep Learning**
  4. systems estimation study performance efficient robust evaluation tracking case improved nonlinear linear **Evaluation methods**
  5. image classification scheme equations prediction online research feature based methods deep nonlinear **Computer Vision**
  6. based analysis optimization dynamic application power applications modeling hybrid methods feature cloud **incoherent**
  7. learning approach algorithm design nonlinear distributed multiple problems machine deep optimization systems **Machine and Deep learning**

8. control sensor selection sensing optimal nonlinear feature distributed systems adaptive tracking networks **Distributed Systems**
9. model wireless stochastic communication networks sensor energy power hybrid nonlinear dynamic based **Wireless Networks**
10. data network adaptive new smart neural based cloud deep analysis dynamic application **Machine and Deep Learning**

5.2 Combined Topic Models

In this part of the exercise we used the external library contextualized-topic-models to solve the task. With help of this library, we could easily build and train the model. The output generated, including the names assigned, is listed below.

- **Before 1990**

  1. system design data analysis using processing computer distributed image digital **Distributed Systems**
  2. information software science management research review development chemical new introduction **Software**
  3. de und von fuumlr zur der des la die et **incoherent**
  4. sets graphs set number classes properties finite boolean degrees types **Graphs**
  5. control systems model optimal linear nonlinear theory estimation identification application **Optimization**
  6. note problem technical letter problems editor sequential solution optimal machines **Optimization**
  7. language programming recognition pattern languages program natural automatic machine approach **Programming Languages**
  8. algorithm algorithms method parallel search efficient using binary computing matrix **Algorithms**
  9. network networks architecture performance simulation protocol computers local digital communications **Communications**
  10. logic propositional symbolic proof semantics calculus logics calculi modal deduction **Logic**

- **From 1990 to 2009**

  1. systems control linear robust stability nonlinear feedback optimal class uncertain **incoherent**
  2. graphs number graph trees complexity sets automata degree groups random **Graphs**
  3. analysis study data molecular models functional human modeling brain dynamics **Models of Human Brains**
  4. underwater feasibility terminal window incorporating handling positioning reactive nonstationary benchmark **Evaluation Methods**
  5. problems problem method solution equations methods numerical optimization order solving **Optimization**
  6. networks wireless mobile network sensor routing protocol performance multicast service **Wireless Networks**
  7. using based image classification recognition neural fuzzy images detection segmentation **Computer Vision**
  8. information special review research web issue introduction technology computer science **Internet**
  9. system design development software decision process support implementation framework distributed **Distributed Systems**
  10. estimation power frequency channel circuit low channels blind array cmos **incoherent**

- **From 2010 onward**

  1. finite equations differential equation approximation fractional solutions problems boundary numerical **incoherent**
  2. systems control feedback stability consensus adaptive output sliding nonlinear multiagent **Distributed Systems**
  3. learning neural deep network machine prediction convolutional classification recognition using **Machine and Deep Learning**
  4. wireless networks sensor allocation protocol vehicular access resource secure radio **Wireless Networks**
  5. cascade multi stage simplified redundancy adjustment buildings train marine window **incoherent**
  6. model fuzzy decision chain approach group making process supply risk **Risk Management**
  7. image segmentation images feature matching color sparse fusion transform based **Computer Vision**
  8. optimization algorithm power system scheduling swarm electric planning multiobjective energy **Optimization**
  9. data land surface temperature water mapping soil satellite china forest **Environment**
  10. review special issue technology challenges role systematic editorial technologies research **Future research**

6. Discussion

In this section, we discuss the different steps taken to preprocess the data used in this exercise. The task makes use of both LDA and CTM. We will first discuss the steps we took in the first part of the exercise (LDA) and then move on to part two (CTM). For the LDA we comment on whether the output we received makes sense, whether or not it is coherrent and what trends we found. In the second part we comment on the performance of the CTM and the similarity of topics and trends.

6.1 Latent Dirichlet Allocation

1. **Does the output make sense?** In our opinion, certain topics above made sense, while others did not. Some topics are just combinations of nouns that make no sense (like Topic 8 in the period before 1990) and others are combinations of keywords from different areas. As an example we can look at Topic 2 in the period From 2010 onward: the ’cloud’ makes the topic look like Internet, but ’Algorithms’ and ’optimal’ makes it look like a theoretical topic about Optimization.
2. **Is the output coherent?** Overall most topics are coherent, but as mentioned above, some topics contain keywords from different areas or make no logical sense.
3. **Which trends did we find?** We found out that before 1990 the topics are mainly basic topics on Computer Science, like Networks or Software. From 1990 to 2009 Computer Vision became more popular. From 2010 onward the topics are mostly about Machine and deep learning, while Computer Vision has still stayed relevant.

6.2 Combined Topic Models

1. **How did the models perform?** In direct comparison, for the current exercise, the CTM generated slightly more coherent topics than LDA. Although some topics are still not coherent enough to summarize and there is an output consisting of stopwords from other languages, the number of topics from CTM we think are incoherent is less than the number of topics from LDA. The keywords generated by CTM appear to be clearer. However we need to mention that the improvement is not significant.
2. **How similar were the topics?**For the period before 1990, the topics are similar. But after that the topics look really different. From 1990 to 2009 the topics of the CTM are widely distributed and have no main theme. The same can be said for the period after 2010.
3. **How similar were the trends?** As mentioned above, the trend of topics generated by CTM is not easily interpretable as the different topics differ widely from each other. The number of topics about Computer Vision and Machine Learning did increase, however it does not seem to be of significance.

7. Conclusion

In conclusion the evaluation of our LDA model led to conflicting results as some outputs made sense while others did not (due to logic and coherence). We determined that before 1990 the topics are mainly basic topics on Computer Science, like Networks or Software. From 1990 to 2009 Computer Vision became more popular. From 2010 onward the topics are mostly about Machine and deep learning, while Computer Vision has still stayed relevant. In direct comparison, the CTM generated slightly more coherent topics than LDA, however the improvement was not significant.