Assignment: Topic Drift

Big Data Analytics

Version: October 22, 2024

1 Introduction

Within a research area, research topics typically shift over time. For example, the raise of big data created new research challenges in the area of data mining, and improved hardware provides novel research opportunities (e.g., multicore and distributed database systems). In this assignment you will try to detect popular topics and how this popularity evolves over time by clustering research publications from a specific research field. More concretely, given a number of publications you can try to determine topics by clustering using a metric based on similarity of words occurring in titles. When you do this over several years you can get a feeling of how topics of interest shift over time.

2 Dataset

We will use the titles of publications within a specific research area from the DBLP dataset. To facilitate the assignment, the dataset for two research areas (Data Mining and Database Systems) is already available on Blackboard as a txt-file, where each row represents a single publication. These publications are extracted from the following conferences (note that some conferences are published as journal articles):

- Data Mining: KDD[1], PKDD[2], ICDM[3], and SDM[4].
- Database Systems: SIGMOD[5][6], VLDB[7][8], EDBT[9], and ICDE[10].

You are free to choose one of these two datasets based on your own interests. Exploring both datasets is not required.

3 Assignment

You should write a program for *(i)* finding topics based on publication titles and *(ii)* use it to show how topics within the field of Data Mining and/or topics in the field of Database Systems evolve over time.

3.1 Part 1: Topic Modeling

In this part of the assignment, you will explore two different approaches to find topics in a collection of text documents (in this case, titles of research publications). The first approach relies on conventional techniques to cluster similar text documents based on frequently occurring words, while the second approach leverages more advanced techniques based on embedding models.

[1] https://dblp.uni-trier.de/db/conf/kdd/index.html
[2] https://dblp.uni-trier.de/db/conf/pkdd/index.html
[3] https://dblp.uni-trier.de/db/conf/icdm/index.html
[4] https://dblp.uni-trier.de/db/conf/sdm/index.html
[5] https://dblp.uni-trier.de/db/conf/sigmod/index.html
[6] https://dblp.uni-trier.de/db/journals/pacmmod/index.html
[7] https://dblp.uni-trier.de/db/conf/vldb/index.html
[8] https://dblp.uni-trier.de/db/journals/pvldb/index.html
[9] https://dblp.uni-trier.de/db/conf/edbt/index.html
[10] https://dblp.uni-trier.de/db/conf/icde/index.html

A conventional approach The steps below give a high-level description of the task at hand.

1. Define a **distance metric** between titles based on the words occurring in the titles. In particular, convert each title into an n-dimensional TF.IDF vector where n is the size of the vocabulary (i.e., the total number of unique words over all titles). For each document j, position i in this vector is the TF.IDF value for term i in document j. To reduce the number of unique words, you might need to preprocess the data (e.g., remove stop words, stemming, etc.).

2. **Cluster** the publications based on the distance metric defined in the previous step. You can use any clustering algorithm you like (e.g., k-means or hierarchical clustering). Tip: make use of existing clustering algorithms in SciPy[11] or Scikit-learn[12].

3. Derive a **description representing the underlying topic** for each cluster. This can for example be done by looking at frequently occurring words in the titles of the publications in the cluster. Keep in mind that some topics span multiple words (e.g., “big data”, “association rule mining” and “graph neural networks”).

Text Embedding Models Text embedding models convert a piece of text into a fixed-size vector representation. These models are typically trained on a large corpus of text and are therefore able to capture semantic information. That is, closely related texts are mapped to similar vectors. Use BERTopic[13] to apply a pre-trained model to the data at hand. Notice that BERTopic combines all three steps above (vector encoding, clustering and topic representation) in one end-to-end pipeline.

3.2 Part 2: Topic Evolution

Visualize the obtained clusters/topics from the previous part and **how they evolve over time**. For example, you can use a line chart where the x-axis represents the year and the y-axis represents the number of publications in a cluster in that year. To smoothen the curve, you can use intervals of multiple years, but allow for some overlap. If the number of clusters is too large, you can manually select a smaller number of relevant clusters for visualization and further exploration.

4 Requirements

4.1 Implementation

You should implement and test your approach on *at least one* of the two provided datasets. Your implementation should cover all steps discussed in Section 3. You are also free to use any existing libraries for the implementation of your approach (for example, libraries implementing the clustering algorithm itself).

Given that the emphasis of the assignment is on exploration, comparing techniques and visualizing results, using a Jupyter notebook is recommended to write code.[14]

4.2 Report

Write a report summarizing the approaches you considered and the main results you obtained. The final report should cover all of the following elements:

- Discuss the different approaches/techniques you considered for each step. Make sure that you cover all relevant aspects (e.g., data preparation and cleaning). If there are relevant techniques you considered but are not implemented, make sure to discuss which ones and why you didn’t use them.

- Give an overview of the external libraries you used. Make sure to cite the sources you used for code, algorithms and ideas.

- Compare the implemented techniques: which one works best and why?

[11] `https://docs.scipy.org/doc/scipy/reference/cluster.html`

[12] `https://scikit-learn.org/stable/modules/clustering.html`

[13] `https://maartengr.github.io/BERTopic/index.html`

[14] Submitting your code as a notebook is *not* a substitute for writing the required report (cf. Section 4.2), even if the notebook is well-documented.

- Explore and compare different values for important parameters (e.g., number of clusters). How do these parameters influence the results?
- Assess the quality of the obtained results: Are the obtained clusters meaningful (i.e., do titles within one cluster indeed cover a common topic)? Are the topic descriptions meaningful? Are the visualizations clear and easy to understand? Including a few examples of clusters and their descriptions as well as some visualizations (e.g. word clouds based on topic descriptions) is highly recommended to illustrate your point.
- Discuss topic drift within the chosen dataset(s) based on your findings: What are the most popular topics? How do they evolve over time? Are there any interesting observations? Make sure to include relevant data and visualizations to support your findings.
- Implementation details: What data set(s) did you use? What is the running time and memory usage? Are there any issues that you encountered? Are there bugs that are still present?
- Conclusion: What are the main findings of your work? What are the current limitations of your approach? Given more time, what other directions would you explore to further increase the quality of detected clusters and topics?

Use diagrams and figures if needed to explain more complicated concepts or algorithms. As a general guideline, The report is expected to be *around 4 to 6 pages long* (including title, figures, references, etc., and assuming a reasonable page layout[15]). Note that this is not a strict requirement. If for example you need more space due to a larger number of figures, feel free to use a couple more pages.

4.3 Submission

You should hand in your code together with the report on Blackboard. Submission of the provided input data and raw output data is *not* required (instead, your report should discuss the main findings based on the output of your implementation, cf. Section 4.2). Make sure to include the name of all group members in your report.

[15] For example, this document uses `\documentclass{article}` and `\usepackage[a4paper,margin=2.5cm]{geometry}`