Bless this Mess

FDS 25/26 Project Report for team Copia_di_Copia_di_Untitled14

Calvano Jacopo
calvano.2082298@studenti.uniroma1.it

Di Timoteo Irene
ditimoteo.2059632@studenti.uniroma1.it

Graziosi Rosamaria
graziosi.2083375@studenti.uniroma1.it

1. Introduction

This project was born in response to the need for improved organization within a file folder. Such a need becomes even more critical when working in groups or downloading large volumes of poorly named files or if you're simply a messy person. The final objective is to construct a deep learning model that, given a folder of text files, organizes it in subject-based sub-folders and detects if there exist different versions of the same document. We propose a multistage pipeline that extracts semantic representations, identifies document relationships and generates meaningful organizational structures. Each stage is modular and can pretty much be independently improved or replaced.

2. Dataset

The dataset used in this work consists of a collection of 350 scientific papers obtained from ArXiv (50 papers for 7 subject). The title and abstract were removed from each document since they were used as queries in the final stages of our pipeline: we wanted to avoid contaminated embeddings. To simulate real-world versioning scenarios, five additional versions were generated for each paper by randomly removing contiguous sections of text. These sections were extracted from different parts of the document, with lengths varying between 10% and 50% of the original content. All documents and their generated variants were stored in a single, nested, JSON file. Based on this preprocessing our ground truth labels of the binary and symmetric relations between documents are the following:

- *version*: a file was generated as a version of the other as explained before,
- *similar*: the documents are from the same subject,
- *unrelated*: all the other cases.

3. Document Embedding

Each document is first transformed into a dense vector representation. To achieve this, a pretrained Siamese neural network encodes the text into a shared embedding space of dimension 512. Siamese NN is an architecture composed of two identical subnetworks sharing the same weights and mapping different inputs. The objective of such architectures is not classification, it's the measurement of semantic similarity between input pairs. Knowing so and having the ground truth in form of labels, we trained a Multilayer Perceptron on top of it. The MLP classifies if document pairs are unrelated, similar or versions of each other. It was also usefull since the adopted Siamese model was pretrained on relatively shorter texts than scientific papers and we weren't sure on how it would behave.

Evaluation. The MLP was trained for a small number of epochs (7) to prevent overfitting due to the limited amount of labeled data and the use of a pretrained block. The model achieved an accuracy exceeding 96% (here), despite a strong class imbalance arising from combinatorial effects in pair generation. To better understand the model behavior, a confusion matrix (here) was analyzed along with precision, recall and F1 score. The results highlighted as critical the misclassifications occurring between the *version* and *similar* classes as also noticeable in a lower F1-score for the *version* category (here). It's critical since the misclassification is almost twice the cardinality of that class.

Table 1. Precision, Recall, and F1-score for each class

Class	Precision	Recall	F1-score
Unrelated	0.997	0.978	0.988
Similar	0.880	0.903	0.891
Version	0.381	0.814	0.519

4. Semantic clustering

Once satisfying embeddings are obtained, a mapping of more complex relationships is computed using a graph-based approach. A similarity graph is built by connecting exclusively documents whose embeddings exceed a cosine

similarity threshold, following a geometrical interpretation of semantic proximity. The resulting graph is then processed using the Louvain algorithm, a topological community detection method. Unlike k-means, this approach does not require fixing the number of clusters and exploits both embedding geometry and graph connectivity density. Each cluster is expected to correspond to a coherent subject area and will be considered as a sub-folder. This intuition and the choice of algorithm are due to this paper: [3] which assure us that the clusters reflect natural topic boundaries, at least better than k-means.

Evaluation. The clustering results were evaluated using both topological and geometric metrics. Conductance was employed as a topological measure to quantify how well a community is isolated from the rest of the graph, where lower values indicate better separation. While the average conductance was satisfactory, the variability across clusters indicates the presence of strongly connected groups that are difficult to further subdivide. Geometric metrics included the Silhouette score and the Davies–Bouldin index, assessing cluster compactness and separation. Both metrics yielded good but not optimal results, which is expected given their purely geometric nature and the hybrid geometric-topological structure of the clustering method.

Table 2. Clustering evaluation metrics (mean $\pm$ std)

Metric	Value
Silhouette score	0.6056 ± 0.0787
Davies–Bouldin index	1.0423 ± 0.1682
Silhouette score	0.2075 ± 0.7800

To compare our performance with results in literature we computed the purity metric. Specifically if we refer to [2] our results are comparable on text dataset that do not differ excessively from the one we chose.

Table 3. Cluster purity across different methods on DBPedia DS

Purity threshold	This model	Vec2GC	K-Medoids
$\geq 50\%$	0.93 ± 0.11	0.94	0.80
$\geq 70\%$	0.52 ± 0.23	0.88	0.54
$\geq 90\%$	0.48 ± 0.22	0.75	0.32

5. Generation of File and Folder Titles

For each detected cluster we try to generate meaningful titles for folders and for individual files using a pretrained large language model, *Qwen*. We used two different prompts: first, a prompt was used to generate concise titles for individual documents. Then, folder names were generated by providing only the documents' new titles, rather than the full document texts. This strategy mitigates input length limitations while preserving semantic coherence. This whole step tries to increase the interpretability and usability of project.

Evaluation. Since we wanted to evaluate how representative these titles are and at the same time retrieve information, we used the next piece of our pipeline as an evaluation system. This approach is loosely inspired by [1].

6. Information Retrieval

In this part of the work, we implemented a semantic Information Retrieval (IR) module, based on the cosine similarity. We used the automatically generated title as the query, after transforming it into an embedding through the encoder of the Siamese model. This means that we do not take the traditional lexical matching approach. Indeed the cosine similarity is between the embedding of the query and the embeddings of the full text of all documents in the dataset. This allows the identification of articles whose content is semantically coherent with our request. Subsequently, the same retrieval scheme is extended by using the embedding of the abstract and then the embedding of the original title as the query. The resulting 6 files were evaluated for each of the three query levels by applying a personalized scoring function called *IreneScore*.

$$\text{IreneScore(d)} = \sum_{d \in D} \mathbf{1}_{\text{correct}}(d) \; + \; 0.5 \sum_{d \in D} \mathbf{1}_{\text{similar}}(d)$$

which assigns a score of 1 if the retrieved document matches the correct one or represents a different version of it, 0.5 if the document is semantically similar, and 0 otherwise.

Table 4. Average retrieval scores for different query types

Query	IreneScore	(%)
Generated Title	3.6672	61.1
Original Title	3.7449	62.4
Abstract	2.3697	39.5

This phase makes it possible to assess whether the generated titles correctly reflect the content of the original documents, providing an external measure of the quality of the generation.

7. Others

Notes. Since our code doesn't download always the same files the metrics may oscillate but minimally. We deleted the title from the beginning of the documents but we suspect, in some papers, it might be referenced later on.

Future develpments. Finally, the system could be extended to support the processing of additional file types (e.g. code files and presentations).We also considered developing a RAG-based system to further improve the IR. This would allow us not only to search for files related to a given query, but also to obtain answers based on the content of our files.

8. Roles

- Calvano Jacopo
  Information Retrieval & Report
- Di Timoteo Irene
  Datset preprocessing, Siamese + MLP training & Demo
- Graziosi Rosamaria
  Clustering, LLM & Presentation

References

[1] Inderjeet Mani. Summarization evaluation: An overview. In *Proceedings of the Second NTCIR Workshop on Research in Chinese and Japanese Text Retrieval and Text Summarization*, Tokyo, Japan, 2001. National Institute of Informatics. 2

[2] Rajesh Rao and Manojit Chakraborty. Vec2gc – a graph based clustering method for text representations. 2021. 2

[3] Shenghui Wang and Rob Koopman. Clustering articles based on semantic similarity. *Scientometrics*, 111(2):1017–1031, 2017. 2