Graph Neural Networks for clustering medical documents

Vittorio Torri, Francesca Ieva

Abstract Clustering is one of the most challenging tasks in the field of Natural Language Processing, due to the high dimensionality of textual data. Different types of document embeddings have been proposed in the past, often based on the transformer neural network architecture. In this work, we propose to exploit a graph-based representation combining it with the recent advancements in the field of graph neural networks. While graph neural networks achieved promising results in document classification, their potential for document clustering has not been explored yet. In particular, we propose an application in the medical domain, where document clustering is of paramount importance due to the large amount of information present in medical documents and the difficulties in labelling them.

Key words: Natural Language Processing, Clustering, Graph Neural Networks, Graph Embeddings, Medical Documents

1 Introduction

Document clustering is one of the most relevant tasks in the field of Natural Language Processing (NLP). The field has seen significant advancements in recent years, particularly with the development of the Transformer architecture and the latest large language models (LLMs) like ChatGPT. Nevertheless, there is still a

Vittorio Torri
MOX - Modelling and Scientific Computing lab, Dipartimento di Matematica, Politecnico di Milano, Milan, Italy e-mail: vittorio.torri@polimi.it

Francesca Ieva
MOX - Modelling and Scientific Computing lab, Dipartimento di Matematica, Politecnico di Milano, Milan, Italy e-mail: francesca.ieva@polimi.it
HDS – Health Data Science Centre, Human Technopole, Milan, Italy

need for tailored models oriented to specific tasks in specific domains, for multiple reasons:

- performances of current LLMs have still margins of improvement and are not always state-of-the-art [5]
- the cost of their use and/or deployment is often unsustainable
- in specific domains, like medicine, where sensitive data have to be processed, it is not legally possible to transmit them to commercial companies

In this work, we propose a new pipeline for document clustering that exploits a graph-based representation of the data that is subsequently embedded with a graph-neural network (GNN) autoencoder. To the best of our knowledge, this is the first work to propose the use of graph neural networks for document clustering.

We apply our pipeline to a case study in the medical domain, where there is a huge need of clustering textual data, due to the large amount of useful information that they store and the lack of labelled datasets [9].

The rest of the paper is organized as follows: in Section 2 we discuss the dataset and the clustering pipeline, in Section 3 we present the results and in Section 4 we summarize our findings.

2 Materials and methods

In this section we present the data for the case study and the clustering pipeline, discussing its main components.

2.1 Data

The data used in this study consists of documents from the Italian section of the E3C Corpus [7]. This dataset is composed of publicly available documents collected from a variety of sources, including case reports from medical journals and texts of exams for medical students. This corpus is multilingual, containing data in five different languages, each with three distinct subsets:

- Layer 1: documents manually annotated with respect to clinical concepts, events and temporal relations among events
- Layer 2: documents semi-automatically annotated with respect to the same entities of Layer 1
- Layer 3: unlabelled documents

In this work, we primarily focused on the documents present in Layer 1 and Layer 2 of the Italian section, consisting of 86 and 174 documents, respectively. This is necessary to assess the performance of our clustering pipeline, as we require ground truth labels. Clustering metrics that don’t rely on labels evaluate clustering quality within a specific embedding space. However, the document embedding itself is a crucial part of our clustering pipeline, requiring evaluation.

This dataset does not have direct labels for document classes/clusters and we had to derive them by exploiting the existing labels for clinical concept entities. In particular, Layer 1 and Layer 2 of the dataset are annotated with entities defined as *clinical concepts* and linked to the corresponding concepts in the UMLS ontology [1]. The UMLS ontology includes various types of relationships among its concepts and we exploited the *parent* relationship to derive the more general topics related to the annotated entities. We annotated each UMLS *parent* concept that we extracted from the annotated *clinical entities* with respect to the main subfields of medicine (e.g.: cardiology, oncology, gastroenterology). Consequently, we assigned each document a label corresponding to the most frequent subfield among the parent concepts of its entities. Considering the dataset's relatively limited size and broad scope, our goal is to identify clusters corresponding to medical fields. These clusters typically contain documents with similar symptoms, exams, or procedures. In the annotated dataset we identified 7 clusters corresponding to different medical areas, covering 213 of the 260 documents. Although only documents from Layer 1 and Layer 2 can be used for the clustering and the evaluation, we also leveraged Layer 3 (consisting of 10209 documents) in the unsupervised training of the graph autoencoder, one of the components of our clustering pipeline.

2.2 Clustering pipeline

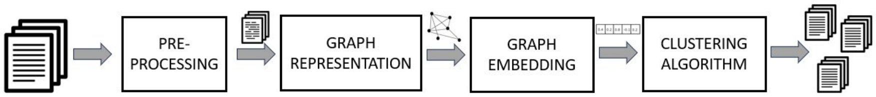

Fig. 1 Schema of the clustering pipeline

Our clustering pipeline is depicted in Figure 1 and consists of four components:

1. Preprocessing
2. Graph-based representation
3. Graph embedding
4. Clustering of graph embeddings

The following subsections detail these components.

2.2.1 Preprocessing

Preprocessing is a fundamental step in NLP pipelines. Given that our pipeline relies on a graph representation aiming to capture the lexical and semantic structure of documents, traditional techniques such as lemmatization and lower-casing were unnecessary. However, we conducted punctuation removal and expanded the most frequent medical term abbreviations present in the dataset.

2.2.2 Graph-based representation

Graph-based representations have a long history in NLP [8], experiencing renewed interest with the emergence of graph neural networks [11]. Various methods exist for representing both individual documents and entire document corpora as graphs. In this study, we propose a document-level graph representation based on dependency parsing, which encapsulates syntactic relationships between words independent of their textual distance. We utilized the Spacy *it_core_news_sm* dependency parsing model for the Italian language [2] to extract these word relationships. Node features were derived from part-of-speech tags, UMLS Semantic Types of words, and Word2Vec embeddings.

2.2.3 Graph embedding

To obtain an embedding vector representing the graph we propose a graph autoencoder architecture. It is derived by the GAE architecture proposed by [3] with the addition of a SAG pooling layer [6]. The basic GAE architecture applies two (or more) layers of Graph Convolution (as defined in [4]), reducing the dimensionality of the node features but not the number of nodes:

$$Z = (\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2})ReLU(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}XW_0)W_1 \tag{1}$$

where $\tilde{A} = A + I$, being A the adjacency matrix of the graph, X the input features, $\tilde{D}$ the degree matrix of $\tilde{A}$ and W the weights of the network.

Our textual data's high dimensionality depends not only on the features of its nodes (tokens) but also on their number. To address this, we incorporate the SAG Pooling layer, which retains only the top k nodes based on an attention score:

$$att\_scores = tanh(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}X\Theta) \tag{2}$$

where Θ the attention parameters. This graph autoencoder is trained using as loss function the binary cross-entropy on the edges of the reconstructed graph, with early stopping on a validation set. We used the Adam optimizer with $lr = 10^{-}5$. The size of the embedding representation is 50, choosen with a grid over number of nodes and length of node embeddings in $[5, 10, 20] \times [5, 10, 20]$.

2.2.4 Clustering

Any traditional clustering algorithm can be in principle applied to graph embeddings, being them dense numerical vectors. We evaluated KMeans, hierarchical agglomerative clustering and HDBScan, finding the best performances in the latter. This confirms previous results on clustering embeddings from neural networks [10].

3 Results

Results are measured in terms of precision, recall and F1-score, weighted averaged with respect to the cluster labels discussed in Section 2.1. In Table 1 we compare the pipeline we are currently proposing with a baseline pipeline based on a TF-IDF representation and with our previously proposed pipeline based on the Umberto model [10]. Our GNN-based pipeline shows improvement over previous models. The Umberto-based pipeline performs even worse than TF-IDF on this dataset. This may be attributed to the longer texts, which would necessitate a different pooling mechanism than the one used in [10]. In Table 2 we compare different clustering algorithm in our GNN-based pipeline, with results highlighting HDBScan as the best.

Table 1 Results of the different clustering pipelines

Pipeline	W. Precision	W. Recall	W. F1-Score
TF-IDF	0.2150	0.3240	0.2343
Umberto	0.0690	0.2626	0.1092
GNN	**0.3027**	**0.3923**	**0.3277**

Table 2 Comparison of different clustering algorithms in our GNN-based pipeline

Cluster. Alg.	W. Precision	W. Recall	W. F1-Score
KMeans	0.1659	0.2961	0.2066
Aggl. Clust.	0.2675	0.3240	0.2351
HDBScan	**0.3027**	**0.3923**	**0.3277**

We executed our pipeline on a Google Colab virtual machine with 8 Intel Xeon @ 2.2Ghz, 51 GB of RAM and an Nvidia T4 GPU. The graph construction required 30 minutes (7 sec/doc), mainly due to the delays in calling the UMLS APIs. The graph autoencoder training required 10 minutes and the subsequent clustering took negliglible time. Considering these numbers, we consider our approach scalable to larger dataset by setting up a local UMLS database instance that might severly reduce the time required for the graph construction.

4 Conclusions

This work introduces a novel pipeline for document clustering, leveraging Graph Neural Networks (GNNs), and applies it to a recent dataset within the medical domain. We also devised a semi-automatic mechanism to derive cluster labels for this dataset, facilitating the evaluation process.

The graph-based representation produced by GNNs combines the power of neural-network based embeddings with a pooling mechanism that reduces the number of nodes in a way that seems to be more effective on long texts with respect to the pooling mechanisms that are typically adopted for BERT-based models. The results compared to other existing methods make this research direction promising. Future work will focus on experimenting with different graph representations and autoencoder architectures.

References

1. Bodenreider, O.:The unified medical language system (UMLS): integrating biomedical terminology. Nucleic acids research 32.suppl_1, D267-D270 (2004)
2. Honnibal, M., Montani, I., Van Landeghem, S., Boyd, A..: spaCy: Industrial-strength Natural Language Processing in Python (2020)
3. Kipf, T. N., Welling, M.: Variational graph auto-encoders. arXiv preprint arXiv:1611.07308 (2016)
4. Kipf, T. N., Welling, M.: Semi-Supervised Classification with Graph Convolutional Networks. International Conference on Learning Representations (2017)
5. Kocoń, J., et al.: ChatGPT: Jack of all trades, master of none. Information Fusion, 101861 (2023)
6. Lee, J., Inyeop, L., Jaewoo, K.: Self-attention graph pooling. International conference on machine learning. PMLR (2019)
7. Magnini, B., Altuna, B., Lavelli, A., Speranza, M., Zanoli, R.: The E3C Project: Collection and Annotation of a Multilingual Corpus of Clinical Cases. In Proceedings of the Seventh Italian Conference on Computational Linguistics, Bologna, Italy. Associazione Italiana di Linguistica Computazionale. (2020)
8. Sonawane, S. S., and Kulkarni, P.A.: Graph based representation and analysis of text document: A survey of techniques. International Journal of computer applications 96.19 (2014)
9. Tayefi, M., Ngo, P., Chomutare, T., Dalianis, H., Salvi, E., Budrionis, A., Godtliebsen, F.: Challenges and opportunities beyond structured data in analysis of electronic health records. Wiley Interdisciplinary Reviews: Computational Statistics, 13(6), e1549. (2021)
10. Torri, V., Ercolanoni, M., Bortolan, F., Leoni, O., Ieva, F.: Clustering Italian medical texts: a case study on referrals. In Proceedings of the Statistics and Data Science Conference (pp. 158-163). Pavia University Press (2023)
11. Wu, L., Chen, Y., Shen, K., Guo, X., Gao, H., Li, S., Pei, G., Long, B.: Graph neural networks for natural language processing: A survey. Foundations and Trends in Machine Learning 16.2, 119-328 (2023)