POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Graph Neural Network pipeline for unsupervised clinical document analysis

Laurea Magistrale in Mathematical Engineering - Ingegneria Matematica

Author: Gabriele Moro

Advisor: Prof. Francesca Ieva

Co-advisor: Dott. Vittorio Torri

Academic year: 2023-2024

1. Introduction

Most clinical documents are present in the form of unstructured data i.e. textual data. Medical documents have an additional level of complexity with respect to the other types of documents, because of the highly technical vocabulary, lack of human labels, and privacy issues of the data. In this thesis work, it will be analyzed the European Clinical Case Corpus (E3C) dataset[1], which contains 10473 clinical cases. This dataset, as it is often the case with medical corpora, does not contain annotations that categorize clearly its documents, which pertain to many different types of diseases. Because of this, it is relevant to develop clustering pipelines able to recognize groups of documents related to patients with similar characteristics, using an unsupervised approach. Inspired by the recent success of the Graph Neural Networks (GNNs) in the field of supervised text classification, we propose a clustering pipeline based on the InfoGraph model [4]. The pipeline exploits a modified InfoGraph GNN to extract a vector representation of the documents, which is subsequently used by traditional clustering algorithms. This is one of the first unsupervised approaches with graph neural networks, and to our knowledge, the first to deal with textual documents. The results show how the vectorial representations of the texts can derive information regarding the origin of the documents but only minimally about the medical content. Several modifications can be implemented to improve the performance of the model, confirming that the field of research under analysis may offer many opportunities for future development. In Section 2 we briefly present an overview of the GNNs model, while in Section 3 we describe the E3C dataset, and in Section 4 we discuss the methodologies. In Section 5 we present the results, in Section 6 we summarize the conclusions of the thesis.

[1] https://github.com/hltfbk/E3C-Corpus

2. Graph Neural Networks

Graph-data structures can be explained as a set of objects that are linked by some sort of connection. Graphs are a universal language for describing complex systems and some examples of graph applications are social networks, chemical molecules, and road maps. Formally, a graph is defined by the couple $G = (E, V)$, where E is the set of edges and V is the set of nodes

(or vertices). The objects are stored as nodes and the relations between them are stored as edges. An edge belonging to E can be written as (i, j), where i and j are the starting and arrival nodes, respectively. A simple but effective representation for graphs is the so-called *adjacency matrix*, a matrix A such that $A[i, j] = 1$ if $(i, j) \in E$, $A[i, j] = 0$ otherwise. The matrix A is symmetric in the case of an undirected graph. Graph datasets, as previously described, have a strong expressive power, and the family of models that have been proven to obtain the best results out of graph datasets are Graph Neural Networks. GNNs use the graph structure and node features to learn a representation vector of a node or the entire graph. GNNs follow a neighborhood aggregation strategy, where the representation of a node is iteratively updated by aggregating representations of its neighbors. After k iterations of aggregation, a node's representation captures the structural information within its k-hop network neighborhood. Each node in the graph collects the information of the neighboring nodes through an aggregation function and combines the information aggregated with the representation of the node at the previous time step. Different learning focuses can be adopted depending on the task: Edge Level, Node Level or Graph Level.
Many GNNs models have been developed in the last decades and they can be divided into 4 categories: RecGNNs, Convolutional GNNs, Graph Autoencoders, Spatial Temporal Graph Neural Networks [5]. Some GNNs models have been tailored for specific settings. For example, when dealing with a scarcity of labels, self-supervised learning models for GNNs are used. Those models exploit a pseudo-labeling schema to learn an enhanced representation of the input. When dealing with NLP tasks, specific GNNs models have been proposed to process textual data, in particular for supervised document classification. Many non-GNNs-based models have been developed for NLP, starting from the classical *Vector Space Models* like TF-IDF and the more recent transformer-based models that with models such as BERT achieved state of the art results in many NLP tasks.

3. Dataset

The E3C is a freely available multilingual corpus (English, French, Italian, Spanish, and Basque) that allows for the linguistic analysis, benchmarking, and training of information extraction systems. In this work, only the Italian subcorpus will be considered, since our goal is to provide a model that would work on Italian texts, specifically. The motivation for which we decided to work only on Italian texts is that language models are generally trained and tested on English texts; this leads to a structural bias when we apply the models to another language. Indeed, Italian and English grammar and syntax are different in many aspects, so models that work well in English may not work well in Italian. The documents are divided into three layers: documents in layer 1 are provided with manual annotations of clinical entities (86 documents), for layer 2 a semiautomatic method is adopted (174 documents); layer 3 instead contains documents without annotations (10213 documents). Each clinical entity in the documents of layers 1 and 2 was assigned to one CUI (Concept Unique Identifier) from the UMLS (Unified Medical Language System).
The documents belonging to the E3C dataset are retrieved from different sources, and therefore they present many differences in length, topic, and form. Overall we can distinguish the sources of the documents between exam documents provided to medicine students with sources such as "Miur", which are generally short documents; journal publications documents coming for example from "Italian Journal of Medicine" and "About Open", which are medium size documents; medicine leaflets coming from "Agenzia Italiana del Farmaco", that are very long documents. Figure 1 shows the boxplot of the length of the documents grouped by source. We observe that the length of the documents is intended as the length of the string representing the text.

4. Methodologies

In this section, we will present the methodologies adopted to cluster E3C medical documents. The steps to obtain the results follow a sequential approach, which is summarized in Figure 2. Starting from the raw text data, the information is processed to get rid of the noisy components of the text, the words are translated

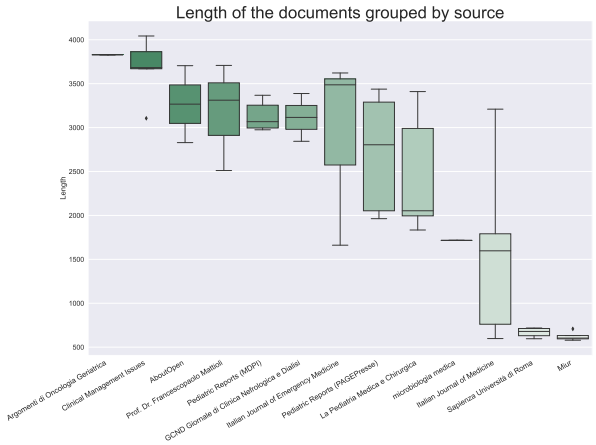

Figure 1: Boxplot showing the length of the documents grouped by source for layer 1. It is evident that "Miur" and "Sapienza Università di Roma" are the sources that correspond to shorter documents.

in a mathematical framework and the texts are represented in a graph format. The preprocessed data texts are then fed to the InfoGraph model which creates a dense representation of each text. The document embeddings are then clustered by means of classical clustering techniques. Lastly, non-ordinary techniques are used to evaluate the goodness of the cluster assignment.

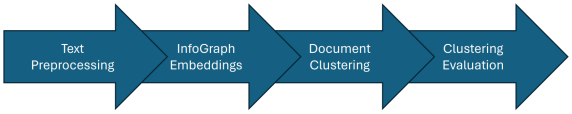

Figure 2: Methodology pipeline.

The text preprocessing part is composed of:

- *data cleaning*: where the text is tokenized, stopwords are removed, and the remaining tokens are stemmed; by doing so the text is filtered of noisy information and should provide more stable results.
- *word embeddings*: which are vector representations of words in a continuous vector space, allowing to compute the similarity between words and process text with machine learning models. We deployed a static version of word embeddings, in particular, the CBOW version of Word2Vec [3].
- *graph construction*: in order to exploit GNNs for document clustering, a conversion to a graph form of the text is required. To do so we followed a learning structure similar to TextLevelGCN [2], where each graph represents a document, nodes represent words and edges represent the proximity relation between words. Figure 3 shows an example of graph structure for a simple sentence.

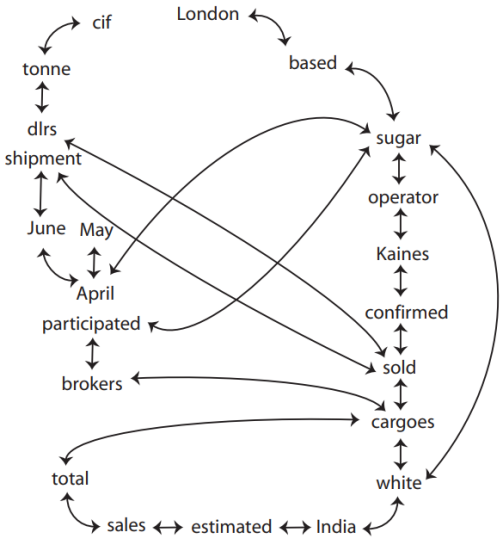

Figure 3: Structure of graph for the sample text *"London-based sugar operator Kaines Ltd confirmed it sold two cargoes of white sugar to India out of an estimated overall sales total of four or five cargoes in which other brokers participated. The sugar, for April/May and April/June shipment, was sold at between 214 and 218 dlrs a tonne cif, it said."*. Picture taken from Hassan and Banea paper [1].

InfoGraph is a self-supervised model, created by Sun et al. [4]. Its goal is to learn the whole representation of graphs in an unsupervised setting. InfoGraph is not a model that is tailored for natural language processing tasks. Also, since E3C doesn't have labels, we don't evaluate the embeddings using a classification downstream task, as explained in the paper, but instead by means of a clustering algorithm. Nevertheless, since InfoGraph has been proven to produce valuable embeddings on graph classification tasks, and considering that embeddings creation with InfoGraph and classification output produced with any classifier (e.g. Logistic Regression, Random Forest, MLP) are two totally separated learning phases, we expected InfoGraph embeddings to perform well also in unsupervised downstream

tasks.
InfoGraph adopts Graph Neural Networks (GNNs) to learn node embeddings for each graph. The node representation learned with neighborhood aggregation of the features will be referred from now on as *patch representation*. From the embeddings of the nodes within a graph, a whole graph embedding is produced by a READOUT function that acts as a summarization of the patch representation into a fixed size graph-level representation, to which we will refer to as *global representation*.
The objective of InfoGraph is to obtain graph representations by maximizing the *Mutual Information (MI)* between graph-level and patch-level representations. In this way, the graph representation can learn to encode aspects of the data that are shared across all substructures. The final patch level and graph level representations are:

$$h_\phi^i = CONCAT(\{h_i^k\}_{i=1}^N). \quad (1)$$

$$H_\phi(G) = \sum_{i=1}^{N} h_\phi^i. \quad (2)$$

h_ϕ^i is the patch level summarized embedding for the node i and $H_\phi(G)$ is the global summarized embedding for the graph G, produced with a READOUT summation function.
The algorithm is presented in Figure 4 using a toy example composed of graph A which has 3 nodes and graph B which has 4 nodes. Both graphs pass into the graph convolutional encoder, to learn patch representations of the nodes and global representations of the graphs. For each node of both graphs, a pair <patch representation, global representation> is given to the discriminator which tries to detect whether the node belongs to the graph or not, by using the labels in a self-supervised fashion. This example shows how negative and positive samples are created to then compute MI maximization. InfoGraph uses a batch-wise fashion to generate all possible positive and negative samples. Considering that only graph A and graph B belong to the batch, there are 7 training inputs given to the discriminator for graph A and 7 training inputs given to the discriminator for graph B. Therefore the discriminator will take 14 pairs <patch representation, global representation> as input for this example. In real case scenarios, the procedure above explained has to be repeated for every pair of nodes and graph present in the batch leading to a much bigger training data for the discriminator.

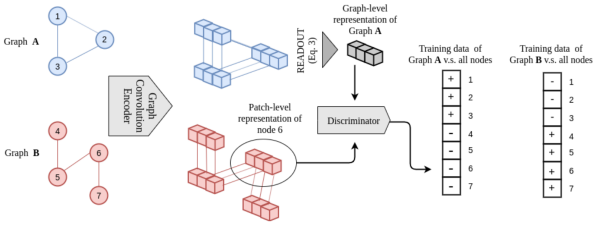

Figure 4: Description of the learning process of InfoGraph. Picture taken from InfoGraph original paper [4].

The produced embeddings are then clustered using *K-means* and *hierarchical agglomerative clustering*. The number of clusters for K-means is decided according to the *Silhouette score*. The visualizations of the results are produced with t-SNE.
Since no groundtruth labels are at the disposal of the E3C corpus, we had to find a way to evaluate the cluster assignment using non-ordinary methods. Classical methods for understanding the goodness of the clustering results such as *Within Sum of Squares (WSS)* or *Silhouette score* are not enough to evaluate the clusters since they don't measure the semantical relations between the document embeddings. The first approach to evaluate the results is UMLS, relying on the clinical annotations of layer 1 as pseudo-labeling. Two UMLS approaches have been attempted: a preexisting package (PyUMLSSimilarity) and a tailored approach using UMLS API to compute the intersection of the broader concepts between different documents as a measure of similarity of the documents. The following evaluation method is *Doc2Vec* which, similarly to InfoGraph, produces embeddings of documents, without relying on graphs but rather on a similar approach to Word2Vec; this method proved to be excellent for both supervised and unsupervised downstream tasks. Lastly, we produced multi-domain labeling for the documents belonging to layer 1 using ChatGPT[2], this last method was fundamental to give an evaluation of the ability of the model to capture the medical meaning of the texts.

[2] `https://chatgpt.com/`

5. Results

In this section, we present the relevant outcome of the explained methodologies. By working on layer 3 embeddings, a visualization analysis with t-SNE showed two distinct clouds of points. Analyzing the documents by source (Figure 5), showed that the bigger cloud was composed only of documents coming from Agenzia Italiana del Farmaco (AIFA). Whereas in the smaller cloud, a less neat separation of documents by source was still present. Since AIFA documents are not properly clinical case texts, we decided to remove them from the analysis retraining InfoGraph embeddings; even in this scenario, the documents appeared separated depending on the source. We proceeded to cluster the remaining documents and by means of UMLS evaluate the results. PyUMLSSimilarity couldn't be applied because of the enormous computational time required, and UMLS API didn't provide reliable results. Before moving on with the embeddings evaluation, we decided to split the documents into two sub-corpa: documents coming from examination texts and coming from journal publications; this separation was applied to reduce the noise created by having documents of many different types, in order to focus more on the medical meaning of the texts. To evaluate the goodness of the clusters on the two sub-corpora, we exploited the ChatGPT labels produced for the 86 documents of layer 1, indeed after clustering all the documents it is possible to check the numerosity of documents of layer 1 with the same domain label which ended up in the same cluster. The results didn't show any significant relation between the medical domain and the clusters, as can be seen in Figure 6, which reports the embeddings of journal publications documents for the four most frequent domain labels (Cardiology, Pediatrics, Gastroenterology, Oncology). Manually checking the documents belonging to the same clusters, led to the same conclusion. To evaluate Doc2Vec embeddings of the E3C corpus dataset, we repeated the analysis with ChatGPT labels done for the document embeddings produced by InfoGraph, as previously presented. While Doc2Vec proved to perform well on many textual datasets, on the E3C dataset it was not able to capture significantly the medical meaning of the documents but it was able to capture relations depending on the different sources of the documents; that is the same conclusion as for InfoGraph embeddings. One reason for these poor results may be the quality of the textual data provided by the E3C dataset or the fact that hidden information within the texts have been considered more important by both unsupervised methods.

To check the last hypothesis and see whether

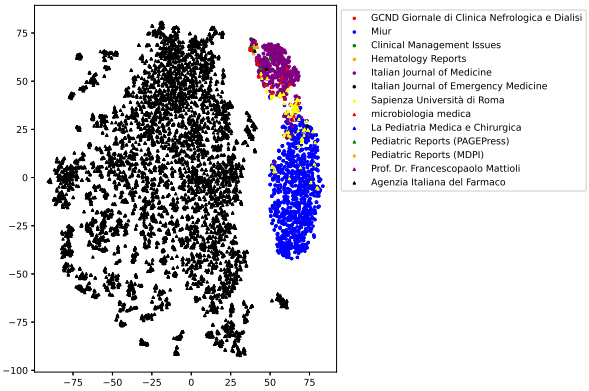

Figure 5: Visualization with t-SNE of the documents embeddings of layer 3 produced by InfoGraph, grouped by source. Documents coming from AIFA form a distinct cloud of points.

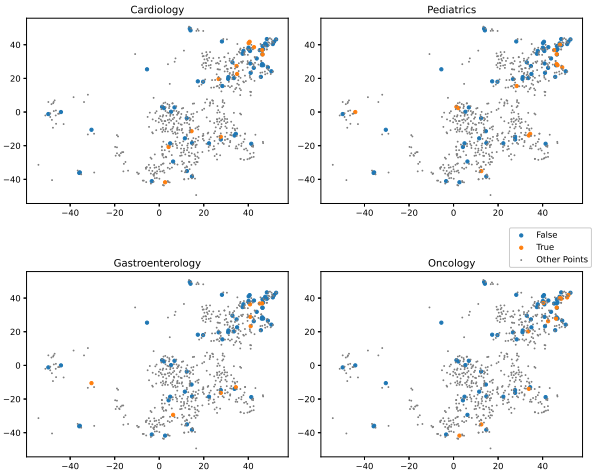

Figure 6: The points assigned as True are the document embeddings belonging to layer 1 with the domain label specified in the title of the sub-plot noted as True; the points assigned as False are the ones with the domain label noted as False. Other Points are the document embeddings of layer 3 which do not have domain labels. The documents represented are only the subset of documents belonging to journal publication sources.

a non-unsupervised approach would be beneficial for the results, a last evaluation method was attempted training a supervised model on the layer 1 documents only, using ChatGPT labels. Layer 1 owns only 86 documents therefore an high-dimensional model like TF-IDF couldn't be applied. To produce fixed, low-dimensional document embeddings we relied again on Doc2Vec. Then, the embeddings were trained following a standard supervised decision tree algorithm in order to classify the medical domain of each text, provided ChatGPT labels. The results are, once again not satisfactory, as shown in Table 1.

Class	Precision	Recall	F1-Score	Support
Cardiology	0.25	0.33	0.29	3
Gastroenterology	0.00	0.00	0.00	2
Oncology	0.33	0.17	0.22	6
Other	0.33	0.44	0.38	9
Pediatrics	0.25	0.17	0.20	6
micro avg	0.27	0.27	0.27	26
macro avg	0.23	0.22	0.22	26
weighted avg	0.28	0.27	0.26	26
samples avg	0.27	0.27	0.27	26

Table 1: Classification report with precision, recall, F1-score, and support for each class. The results refer to the test set.

From the computational side, we ran InfoGraph with a laptop without GPUs (11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz, 32,0 GB ram), this confirms that even if the learning structure of InfoGraph requires many computations (for every node in every graph in the batch mutual information is computed against all the graphs in the batch) it can be considered an accessible model. Highlighting its ability to run effectively on standard hardware configurations, such as a normal laptop, without needing specialized cloud computing or GPU resources, making InfoGraph a viable option for researchers and practitioners who may not have access to high-performance computing infrastructure. The default hyperparameters used for InfoGraph training are Learning Rate = 0.001, Hidden Dimension = 16, Number of Graph Convolutional Layers = 3, and Number of epochs = 10. The chosen number of epochs is low because of early stopping, indeed, after a few epochs, the model stops decreasing the loss function.

6. Conclusions

Inspired by the fact that approaches based on Graph Neural Networks achieved state of the art results in supervised text classification, we developed a pipeline for clustering clinical case documents based on graph neural networks. The pipeline was applied to a medical dataset of Italian texts (E3C). The method is a first attempt to exploit GNNs for document clustering. According to several evaluation methods, the medical meaning of the documents was not captured by InfoGraph. On the other hand, documents coming from different sources have been recognized with similar embeddings. We believe that our work could pave the way for new research developments in this field, for example by changing the evaluation dataset, choosing other GNN-based methods, and enhancing the graph structure to represent the text.

References

[1] Samer Hassan, Rada Mihalcea, and Carmen Banea. Random walk term weighting for improved text classification. International Journal of Semantic Computing, 1(04):421–439, 2007.

[2] Lianzhe Huang, Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng WANG. Text level graph neural network for text classification, 2019.

[3] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 26, 2013.

[4] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. arXiv preprint arXiv:1908.01000, 2019.

[5] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems, 32(1):4–24, 2020.