Procesamiento del Lenguaje Natural, Revista nº 72, marzo de 2024, pp. 137-148 recibido 15-12-2023 revisado 30-01-2024 aceptado 04-02-2024

OntoLM: Integrating Knowledge Bases and Language Models for classification in the medical domain

OntoLM: Integrando bases de conocimiento y modelos de lenguaje para clasificación en dominio médico

Fabio Yáñez-Romero[1], Andres Montoyo[2], Rafael Muñoz[2], Yoan Gutiérrez[2], Armando Suárez[2]

[1]University Institute for Computer Research, University of Alicante.
[2]Department of Computing and Information Systems, University of Alicante.

fabio.yanez@ua.es, montoyo@dlsi.ua.es, rafael@dlsi.ua.es, ygutierrez@dlsi.ua.es, armando@dlsi.ua.es

Abstract: Large language models have shown impressive performance in Natural Language Processing tasks, but their black box characteristics render the explainability of the model's decision difficult to achieve and the integration of semantic knowledge. There has been a growing interest in combining external knowledge sources with language models to address these drawbacks. This paper, *OntoLM*, proposes a novel architecture combining an ontology with a pre-trained language model to classify biomedical entities in text. This approach involves constructing and processing graphs from ontologies and then using a graph neural network to contextualize each entity. Next, the language model and the graph neural network output are combined into a final classifier. Results show that *OntoLM* improves the classification of entities in medical texts using a set of categories obtained from the Unified Medical Language System. We can create more traceable natural language processing architectures using ontology graphs and graph neural networks.
Keywords: External Knowledge, Ontologies, Large Language Models, Graph Neural Networks.

Resumen: Los grandes modelos de lenguaje han mostrado un rendimiento impresionante en tareas de Procesamiento del Lenguaje Natural, pero su condición de caja negra hace difícil explicar las decisiones del modelo e integrar conocimiento semántico. Existe un interés creciente en combinar fuentes de conocimiento externas con LLMs para solventar estos inconvenientes. En este artículo, proponemos *OntoLM*, una arquitectura novedosa que combina una ontología con un modelo de lenguaje pre-entrenado para clasificar entidades biomédicas en texto. El enfoque propuesto consiste en construir y procesar grafos provenientes de una ontología utilizando una red neuronal de grafos para contextualizar cada entidad. A continuación, combinamos los resutlados del modelo de lenguaje y la red neuronal de grafos en un clasificador final. Los resultados muestran que *OntoLM* mejora la clasificación de entidades en textos médicos utilizando un conjunto de categorías obtenidas de Unified Medical Language System. Utilizando grafos de ontologías y redes neuronales de grafos podemos crear arquitecturas de procesamiento de lenguaje natural más rastreables.
Palabras clave: Conocimiento Externo, Ontologías, Grandes Modelos de lenguaje, Redes Neuronales de Grafos.

ISSN 1135-5948 DOI 10.26342/2024-72-10

1 Introduction

This work is centred on the premise that using structured external knowledge can help during the fine-tuning process of large language models, and it also makes the architecture more traceable and explainable as it provides semantic knowledge during the process. To validate the premise, a multilabel classification task is chosen. In this task structured knowledge is used with language models forming an even larger architecture which combines the language model with a graph neural network (GNN) in a final classifier.

This work aims to insert structured external knowledge into the decision-making of a model based on pretrained language models, improving the results obtained in classification tasks and obtaining a final architecture (*OntoLM*) that will allow traceability through the GNN and the initial structures obtained from UMLS.

An ontology defines the possible relations between different types of entities and is used as a schema to decide how relational information should be stored in an ordered way. The rules defined in the ontology are expressed in the final knowledge base (KB) derived from this ontology. KBs store information about many domains in a structured way. Big KBs like UMLS or WordNet have proven their usefulness in many downstream tasks where factual information is needed, reducing the amount of wrong information returned (Chen et al., 2017). AlKhamissi et al. (2022) consider the following criteria as the most important characteristics for considering a language model as a KB:

- **Accessibility**: all the information of a KB can be queried directly.
- **Easy to edit**: every entity or relation can be modified with minor effort.
- **Consistency**: queries with the same meaning should give the same result.
- **Reasonableness**: how suitable is the application of reasoning techniques over their structure rather than deep learning models.
- **Explainability and interoperability**: explainable algorithms and techniques are more suitable; for example, knowledge base schema or path walking techniques.

By contrast, big deep learning models used in Natural Language Processing (NLP) store large amounts of information through their training with large amounts of text, as shown in the most recent cases with BERT (Devlin et al., 2019) or GPT-4 (OpenAI et al., 2024). Language models have proven to be very useful in numerous tasks carried out in language processing. Their different architectures allow them to cover both classification and text generation tasks. However, these models have a large amount of probabilistic knowledge which cannot be interpreted.

Using only language models can present problems because of the lack of internal reasoning in such a model, as well as biases (Bender et al., 2021) and toxic information (Gehman et al., 2020) contained in them. The information obtained in these models is not easy to update, so they tend to be easily outdated due to the high cost of re-training them. Also, these models have many inconsistencies, as shown by works that obtain different information using prompt engineering techniques (Elazar et al., 2021).

Finally, traceability, interpretability, and explainability are easier to achieve with a well-defined ontology that generates information based on certain rules or schema and their graph structure (Agarwal et al., 2023). Deep learning models that consider the entire structure of a graph in the training data often provide more traceable structures that can be understood intuitively (Zhou et al., 2020).

The paper is structured as follows: Section 2 discusses other works using similar approaches, trying to provide semantic knowledge with external knowledge or training data for language models. Section 3 describes the aim and characteristics of the corpus associated with the experiment. The next sections, 4 focuses on the whole architecture of the experiment with a specific focus on data processing and model training 5. Subsequently, the results are reported in section 6. The discussion of the results obtained is carried out in section 7, whereas conclusions and future work arising from the discussion are explained in section 8.

2 Related Work

The use of external knowledge and language models has been extensively researched to address the issues encountered in language models. Some works, such as (Kaur et al.,

2022) or (Sun et al., 2021), aim to pretrain the model by incorporating semantic knowledge or altering the existing architecture. Others train a language model from scratch using an innovative masking approach (Zhang et al., 2019), improving many benchmarks. In both cases, the computational cost is high, making it difficult to adopt similar experiment strategies.

Other approaches try to bring semantic knowledge into the language model without updating the language model parameters, either by using pre-processing (Sun et al., 2024) or post-processing (He, Zhang, and Roth, 2022) techniques. These approaches are usually less expensive, making them more accessible and versatile than previous examples.

The factual knowledge contributed to the language model can have an unstructured origin, as in the case of Peng et al. (2023), or it can come from structured knowledge bases, where the information is mainly organised in the form of triples (Huang et al., 2022).

The advantage of using a structured knowledge source as external knowledge is the elimination of ambiguities present in the text, as well as an ordered information structure that does not introduce more noise than necessary and the provision of semantic knowledge, such as synonymy, hyponymy, and hyperonymy or antonymy relations (Mrkšić et al., 2016) depending on the knowledge base used.

Previous work has attempted to provide structured knowledge by capturing the semantics of KBs and feeding this knowledge into deep learning models from modules specialised in this task (Piad-Morffis et al., 2019).

Other works have been carried out that attempt to benefit from the knowledge present in knowledge bases with GNNs because of the inherent relation of this architecture with their different nodes. Jiang et al. (2020) proposes to use knowledge from a graph to perform text classification, in their case they create the graph by performing Named Entity Recognition (NER) on short texts, augmenting the information obtained with a general knowledge base and initiating embeddings of each entity using Word2Vec (Mikolov et al., 2013). The graph is processed using Gated Graph Neural Networks (GGNNs) (Li et al., 2017), and they also process the whole text with a pre-trained language model (PTLM). Finally, they use an attention background on the GGNN and the PLM results to classify the text. There are other methods to create embeddings of the entities and relations of a knowledge graph. These methods can be considered contextualised embeddings from knowledge graphs (Yáñez Romero et al., 2023-09).

Another example is Feng et al. (2020), who use pre-trained language models and knowledge from different ontologies to answer questions with a fixed number of answers as context. In their work, they form different graphs with ontology entities from the entities detected in both question and possible answers. This information is passed through a GNN that considers the type of relation between each node and a node scoring system to filter the possible paths between questions and answers. This novel way of applying external knowledge to language models has a major problem: it is used specifically to respond to questions with a fixed number of answers.

In this proposal, an architecture similar to (Yasunaga et al., 2021) is used to classify entities detected in a given text. For this purpose, language models trained in the specific domain of the text and ontologies with knowledge of the same domain will be used. The architecture of the graphs used during training will be adapted to the proposed objective, and the GNN introduced by Feng et al. (2020) and the improvements introduced by Yasunaga et al. (2021) will be utilized.

3 Corpus

The corpus used to classify medical entities has been created by annotating medical terms found in abstracts of papers obtained from PubMed. Annotated texts focus on diseases, as these texts were collected to classify entities related to diseases and ailments. The annotations made contemplate 40 different categories obtained from the semantic types of UMLS. Specifically, this corpus has been annotated semi-automatically by performing NER on the abstracts using NER models obtained from sci-spacy (Neumann et al., 2019), namely 'en-core-sci-lg'. Then, each annotation was supervised, and labels that did not correspond to the context of the entity were removed. However, the corpus used is very unbalanced, as differences of 1 to 100 can be found between the categories with the lowest representation and those with the highest representation. This problem was mitigated by undersampling the dataset.

Finally, a corpus was obtained where each entity can be annotated with more than one category since, in its context, this entity can be considered within different UMLS categories, e.g. in Text 1 the entity *pharmacological treatments* can be classified as *healthcare activity* or *research activity*. Therefore, we are faced with a multi-label classification problem.

> *This work analyzed salivary Lf concentration under different handling conditions and donor-dependent factors, including age, inter-diurnal variations, physical activity, and* ***pharmacological treatments****.* (1)

However, in the corpus used, most entities are classified with only one label, with a few examples having two labels and almost none with more, as shown in Figure 1.

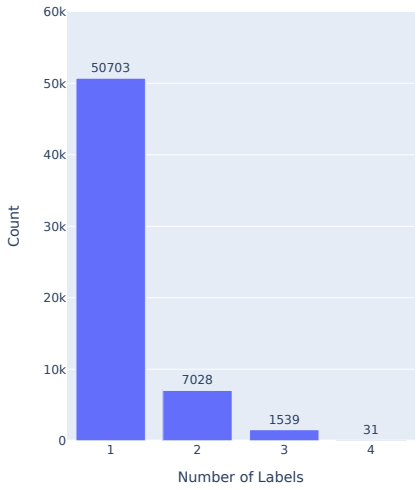

Figure 1: Clustering of training data based on the number of labels.

4 *OntoLM*

The proposal performs a supervised classification task on annotated medical entities obtained from biomedical texts. In brief, supervised learning is performed by augmenting a pre-trained language model with the knowledge from the UMLS medical ontology, carrying out the training of a GNN. The different sections of the architecture are listed as follows:

1. The starting point is the data obtained from the corpus with their corresponding labels.
2. The UMLS ontology database is then used to augment the annotated data with the possible entities detected in UMLS (4.1).
3. The graphs necessary for the model's training are created according to section 4.2.
4. The language models used to represent the entities and process the text are named in section 4.3.
5. Finally, a summary of the task covered in the experiment is given 4.5.

Figure 2 shows the complete proposed architecture, starting from the texts with annotated entities. In the pre-processing stage, each text document is represented in light yellow, the database and the entities extracted from it can be seen in orange, and the input tensors to the model are in grey. During the training step, depicted in light blue/salmon, the model's components are frozen/learning, respectively. Numerical values represent the order of the data flow.

The complete experiment considers all three components: GNN, LLM, and a multilayer perceptron (MLP).

4.1 Ontology structure

Understanding the structure of the initial ontology from which one starts is essential to forming a coherent graph for the proposed task. In this case, UMLS will be used as it contains much knowledge from the medical field (Bodenreider, 2004).

Considering UMLS entities and relations as a graph, the minimum structure of this database would be triples. Each triple consists of a head entity e_h and a tail entity e_t with its respective relations r so that a triple would be represented by the expression (e_h, r, e_t). It is possible to generate a knowledge graph from UMLS using specific data tables that indicate the relations between the different entities, i.e. triples.

These concepts and relations constitute the main data source of UMLS, known as its Metathesaurus. However, UMLS has other sources of knowledge, such as the Lexicon, which generates the different linguistic variants of a term, or the semantic network, which generates higher-level categories that encompass the concepts in the database. The

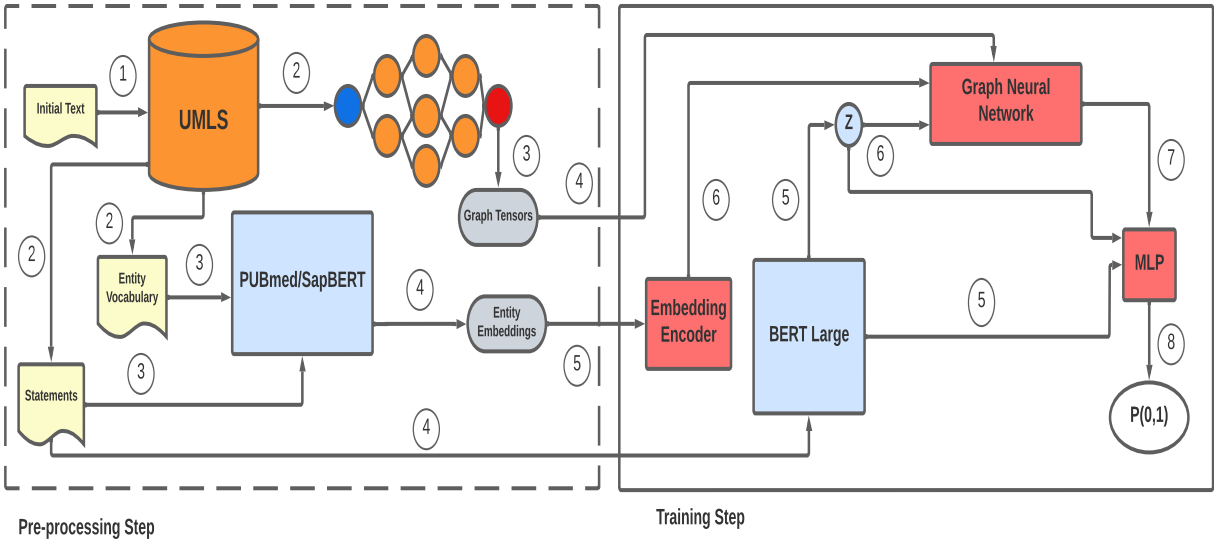

Figure 2: OntoLM architecture. The box above shows all the pre-processing of the data carried out before training. The bottom box shows the training stage.

UMLS semantic network classifies each concept based on Semantic Types (TUIs) (McCray, 1989), which can represent direct relations between the different concepts or classify these concepts in higher-level categories. Moreover, TUIs are organised hierarchically among themselves and can have a subset within another set.

The TUIs used to catalogue the different concepts can be used as categories in entity classification problems within the biomedical field. Considering all the classification TUIs, there are 127 different ones, forming a large number to be used in classification problems. Of the 127 initial categories obtained from UMLS, 34 were eliminated because they did not provide value for classifying disease-related entities, such as *Temporal concept* or *Geographical Area*. This leaves a total of 93 categories, a particularly high number for multilabel classification using language models. For this reason, the 93 categories have been reduced by grouping them by their hierarchical relations so that *Plant*, *Fungus* or *Animal* can be grouped under TUI *Organism*. The number of categories considered has been reduced to the 20 most representative ones to balance the final categories obtained. The 40 initial categories and the chosen 20 categories are shown in Table 1.

For each example to be classified, 20 graphs are generated with the detected entities of the text, which a GNN then processes. Also, 20 statements are generated and processed by the language model. A graph and a statement are generated for each possible category among all those considered. The construction method of each graph and statement is indicated in the following sections.

UMLS Id	Category Names	Nº Labels
T001	Organism	-
T005	Virus	-
T007	Bacterium	-
T018	Embryonic Structure	-
T023	**Body Part Organ Or Organ Component**	**801**
T025	**Cell**	**801**
T026	**Cell Component**	**801**
T028	**Gene Or Genome**	**801**
T032	Organism Attribute	-
T033	**Finding**	**801**
T037	Injury Or Poisoning	-
T038	**Biologic Function**	**801**
T043	**Cell Function**	**801**
T046	**Pathologic Function**	**801**
T047	**Disease or Syndrome**	**801**
T049	**Cell or Molecular Dysfunction**	**801**
T050	Experimental Model of Disease	-
T055	Individual Behavior	-
T058	**HealthCare Activity**	**801**
T062	**Research Activity**	**801**
T066	Machine Activity	-
T069	Environmental Effect of Humans	-
T070	Natural Phenomenon or Process	-
T073	Manufactured Object	-
T079	**Temporal Concept**	**801**
T085	Molecular Sequence	-
T091	Biomedical Occupation Or Discipline	-
T093	HealthCare Related Organization	-
T098	**Population Group**	**801**
T101	**Patient or Disabled Group**	**801**
T103	**Chemical**	**801**
T114	Nucleic Acid Nucleoside or Nucleotide	-
T116	**AminoAcid Peptide or Protein**	**801**
T121	**Pharmacologic Substance**	**801**
T123	**Biologically Active Substance**	**801**
T167	Substance	-
T184	Sign or Symptom	-
T190	Anatomical Abnormality	-
T201	**Clinical Attribute**	**801**
T204	Eukaryote	-
	Total Statements	**15321**

Table 1: The 40 initial categories considered in the classification task, and the 20 final categories used after undersampling the dataset.

4.2 Proposed Graph Structure

For the classification of words from a text, it is necessary to modify the network architecture proposed in (Feng et al., 2020) and (Yasunaga et al., 2021), since it is not about answering questions. Therefore the possible answers cannot be used as a context.

In this structure, there is an initial entity, which is the target entity to classify, and the rest of the entities detected in the text belonging to the biomedical field. The other entities detected will serve as context to classify the target word.

To introduce the context of the entity to be classified and the possible classification it refers to, entities that do not exist in UMLS are created representing the exact word found in the text, and new relations that will connect these entities with UMLS entities. For instance, the word 'results' in a medical text may refer to different entities within the knowledge base, such as 'Clinical results' or 'Experimental results'.

From the initial entity, which is the annotated entity, using matching with n-grams of three characters, the possible entities referred to by that word are obtained, each with their respective classifications. The initial entity is related to these entities from the ontology using a newly created relation *meaning of*. From the ontology entities, the rest of the entities directly connected to them that share one or more semantic types can be obtained using the UMLS database. This step can be done many times, increasing the size of the final graph. The final node of the network is each of the possible categories used in the architecture. This node will be directly connected to the rest of the entities detected in the target text related to the category (based on UMLS possible entities). The new relation used in this case will be *belongs to*. The considered relations can be expanded with direct relations to the possible TUIs of the word to be classified, further extending the graph and thus connecting to the initial entities. The intermediate triples obtained from the ontology present the different relations considered in the UMLS version.

Figure 3 illustrates the architecture of each graph; every node represents an entity, the blue nodes being the target entity in the text to be classified. The green nodes are the possible nodes obtained from the ontology. The orange nodes represent the remaining nodes obtained from the text that have a category that coincides with the possible categories of the green nodes (based on three characters n-gram matching on UMLS). The red nodes represent one of the 20 possible categories, which matches the orange nodes category. Finally, the white nodes represent those obtained from existing relations in UMLS with the rest of the previously mentioned nodes.

To avoid information loss and improve the results if many jumps are made, the contextual node Z is used (Yasunaga et al., 2021). This node connects the initial node (target word to be classified) with the final node (possible category).

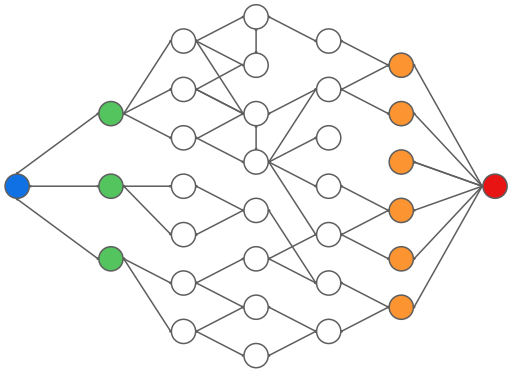

Figure 3: Graph structure proposed by each statement.

In cases where the target category is not related to the entities of the text, only the category is added as an isolated node. The other categories of the text are not added. Thus, a network is formed by the nodes obtained from the initial entity and the isolated node (connected only by the context node Z).

The proposed architecture is processed using the GNN introduced by Feng et al. (2020), then a pooling is performed on the GNN and fed into an MLP together with the language model data and the context node Z.

4.3 Language Models

To represent the nodes of each graph in a format compatible with the GNN, it is necessary to use embeddings containing the information of each entity. Each node is initialised using a specific language model for this task in this case.

Language models can be used to create entity embeddings, as they store a large amount of knowledge in their model weights. With

this in mind, language models trained on a specific domain can represent entities and their relations from that domain for subsequent tasks. This is the case of other works such as (Wang et al., 2023) or (Wang et al., 2022), where entity embeddings are created using language models for entity linking and relation inference.

A BERT model trained with UMLS data is used to generate the embeddings of the medical entities. This model was trained to represent the different names that the same medical concept can have in a similar way, which is ideal for the present task (Liu et al., 2021). In addition, the SapBert model used is based on PubMedBERT (Gu et al., 2021), a BERT model pre-trained in the biomedical domain, specifically taking texts from PubMed. In this way, vectors of each biomedical entity are obtained, giving as input each of the biomedical concepts obtained from the graphs in a text format to the tokenizer.

In the architecture, the BERT Large pre-trained language model is used. This model will receive each of the statements indicated in the following section as input data.

4.4 Language Model Statements

The input to the language model associated with the node is the entire context of the text in question, such as Text 2.

$$[CLS] + Sentence + [SEP] + term + [SEP] + Label \quad (2)$$

Where [CLS] and [SEP] are the special classification and separation tokens used in BERT, respectively. Considering Text 1, we would have has input for term *pharmacological treatments*, labels *healthcare activity* and *research activity*, having two different inputs for the LLM.

The information obtained from the graphs after using the GNN proposed by Yasunaga et al. (2021) is combined with the output of the language model, representing that graph along with the contextual node obtained from the language model but adapted to the size of the GNN nodes. The pre-trained language model will return an embedding size equal to its last hidden layer.

4.5 Classification problem

The proposed classification problem will try to classify each entity detected in the target text among the 20 reduced categories obtained from the UMLS semantic types. The proposed architecture as in Yasunaga et al. (2021) employs an MLP at the end of the architecture. This MLP receives as input data the pooling vector obtained from the GNN, the output of each statement of the language model, and the vector that represents the context node Z. This concatenation will be received a total of 20 times, 1 for each category considered and will return a single probability that will be compared with the label in question.

The classification problem considered is multilabel, so each word to be classified can have more than one associated category, and in this case, no category is mutually exclusive. To carry out the classification, a sigmoid function and then binary cross entropy are used as the final activation function of the MLP, comparing each result obtained by the concatenation of a statement, graph, and context node vectors with the label in question.

The loss function considered is defined at the end of the MLP, so back-propagation updates the weights of the MLP and the GNN, as well as the linear transformations carried out to adjust the vectors representing each node of the graph to the dimensions of the GNN. The language model weights are kept frozen ($OntoLM_F$) or unfrozen ($OntoLM$) depending on the experiment.

5 Experimentation

The data obtained from the corpus are not correctly balanced, e.g. the category with the highest representation has 100 times more examples than the category with the lowest representation. This leads to performing an undersampling task on the data before training the model. Multi-Hop Graph Relation Network (MHGRN) introduced by Feng et al. (2020) also considers the number of different relations, but in previous question-answering experiments, the number of different relations is not large. In this case, the experiments consider all relations extracted from UMLS.

5.1 Undersampling

An undersampling task was carried out during the experiments to balance the dataset used. Balancing the dataset considerably improves the results obtained, since otherwise good results are only obtained with the labels with the highest representation. The final dataset used has a total of 800 instances for each of the labels, and each of these instances can have more than one label. During the undersam-

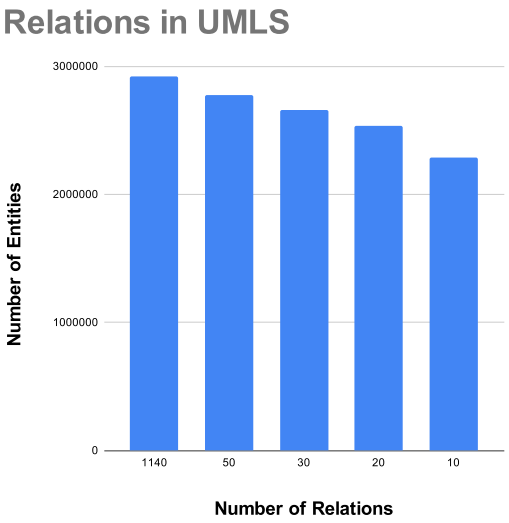

Figure 4: Number of different relations and the total coverage of entities using these relations, around 95 percent of the database uses only 50 relations.

pling task, training data was removed once the limit set per label was reached. Having a maximum of 800 in the most representative cases. Oversampling the data that has very few instances is not recommended since the training of the model gives very poor results on the categories where oversampling is performed. Therefore, removing categories with less than 800 instances is the best solution, reducing the dataset to 20 categories. The number of instances for each category is shown in Table 1.

5.2 Number of relations

All the relations present in the UMLS version have been used; however, of all these relations, a few have much higher representation reaching 95 per cent of the whole database downloaded with the top 50 relations, not counting the introduced relations *belongs to* and *meaning of*. With this in mind, a large part of the database can be represented with few relations, which is likely to positively affect the classification task by reducing the training complexity of the GNN. The representation of the database considering the number of relations can be seen in Figure 4. In this case, experiments using a simplified database with reduced relations will be conducted in future works.

5.3 Baseline

To carry out the experiment, the pre-trained BERT model is considered as baseline, specifically the large version obtained from the HuggingFace library together with an MLP comprising two hidden layers for final classification. This language model is considered the baseline since the whole system will use this model and the rest of the proposed architecture, to perform the classification.

6 Results

Precision and recall have been measured for each category considered during the experiment. Specifically, confusion matrices were used for each category, thus obtaining true positives, true negatives, false positives, and false negatives. In this way, the F1 score of each category was obtained, and the overall results can be seen in Table 2. Figure 5 shows the best micro results obtained for the model with better macro F1 ($OntoLM_F$).

Model	Accuracy	Precision	Recall	F1
Baseline	0.97	0.42	0.83	0.56
OntoLM	0.96	0.59	0.62	0.60
OntoLM_F	0.97	0.74	0.62	0.68

Table 2: Macro Accuracy, Precision, Recall and F1 for each experiment. Results for the best epoch.

7 Discussion

The proposed final architecture trains 1.2 million parameters, 300 times less than pre-trained language models such as BERT Large. However, the training becomes computationally expensive due to the large number of tensors representing graphs used as model input data compared to classical language model training, which employs only text tensors during this stage. The experiments were carried out using one 40 GB A100 GPU, spending a total of 12, 18 and 3 hours for training three epochs on *Baseline*, *OntoLM* and $OntoLM_F$, respectively and incrementing the batch size as much as possible to fill the GPU memory. Moreover, considering a graph and a statement for each possible term category increases the computational cost considerably. An attempt has been made to reduce the computational cost of the input data by reducing the size of the graphs, since in the case of the experiments carried out by Yasunaga et al. (2021), the size of the graphs used is 200

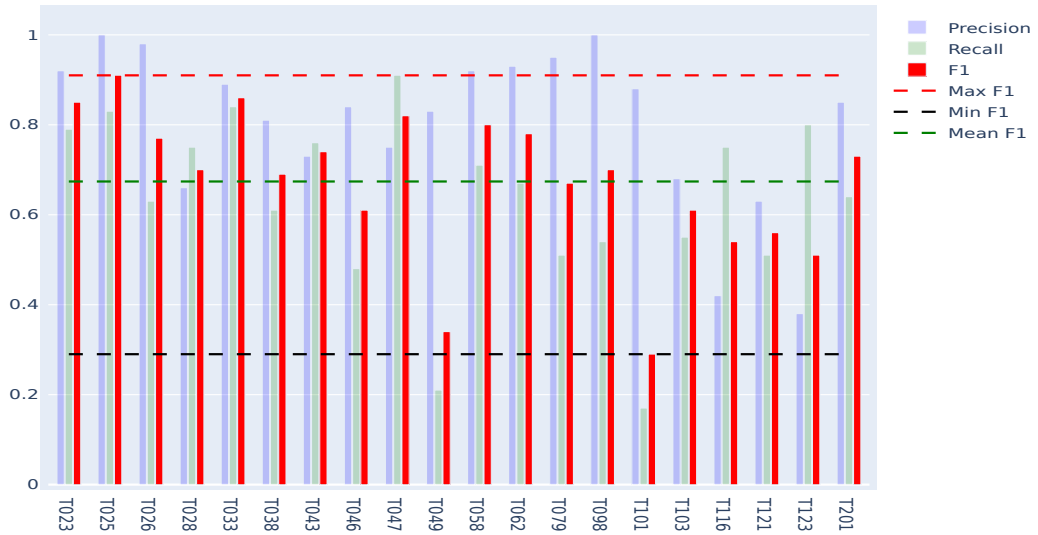

Figure 5: Precision, Recall and F1 for the best experiment in each of the 20 categories used.

nodes at most. In our particular case, reducing the size of the graphs to 100 nodes at most allows us to obtain good representations of each instance while reducing the weight of the input tensors by 40 per cent.

During the evaluation of the training epochs, notably during the different experiments, the real positives were not learned in the first two epochs. This is associated with the fact that this problem is a multilabel classification with too many negative categories, i.e., most of the label categories are zero, so returning zero in all categories for each training instance reduces the loss function considerably. Next, to reduce the loss function, it is necessary to identify the true positives in the output.

Table 2 shows that significant improvements have been obtained when using the proposed architecture over the baseline. However, the recall obtained in the baseline is far superior to the OntoLM experiments, suggesting that proper hyperparameter tuning is likely to give better results when running the full architecture. Running experiments with the full architecture yields better results by keeping the language model frozen ($OntoLM_F$), suggesting that the GNN architecture better adapts the knowledge of the language model for downstream tasks compared to the unfrozen language model ($OntoLM$). This result suggests that the proposed architecture can serve as an alternative to fine-tuning or that we can improve the results obtained by initially performing traditional fine-tuning on the language model and then attaching it to the overall architecture by training the GNN. As an alternative to fine-tuning, the proposal presented in this work is valid as, in addition to the better results, the computational cost (both in time and resources) is considerably reduced if the language model is kept frozen. The code of the experiments is available on GitHub [1].

During the realisation of each experiment, notably in the first two training epochs, the models do not classify any statement as positive, thereby obtaining only true and false negatives. The architecture finds as a first valid option to optimise all results in this way to reduce the loss function. Then, if the learning rate is low enough to classify the true positives, each model will learn to classify them, obtaining the best results in the first 10 epochs. This is quite likely considering that the labels used have very few positive categories, with 1 or 2 out of 20 in most cases.

Finally, the initial embeddings of each graph are not as expressive as they could be, mainly because the relations between the different nodes are not represented with contextualised embeddings from the beginning as with other methods. It is worth testing in future work by initialising these nodes with contextual embeddings based on their respective ontology and modifying the GNN architecture to process those contextualized embeddings.

[1] https://github.com/FabioDataGeek/OntoLM

8 Conclussion and Future Work

Given the results, multilabel classification tasks are improved by incorporating external structured knowledge. As far as we know, few works have performed the classification task with such a high number of categories. In the case of (Lee, Lee, and Ahn, 2022) they use 45 categories to perform multilabel classification of texts. However, in our case, the objective is not to classify the text but the possible entities found in a text from a certain domain. To the author's knowledge, very few works perform this specific task with so many categories. However, in tasks such as classification based on International Classification of Diseases codes, 10th edition (ICD-10), within the biomedical field (Gérardin et al., 2022) both entity and text classification studies exist, a task that is especially relevant to the purpose of this work.

Experiments show us an alternative way of adapting a language model to a specific domain without changing the domain weights, which is less computationally expensive and faster than loading the language models for fine-tuning. However, the time spent preprocessing the data to generate each graph must also be considered. The results obtained with the proposed architecture open up several lines of research, including the following:

1. The combination of ontologies with language models in other domains to perform classification tasks. Using this architecture with other ontologies can be especially useful to cover other NLP tasks such as word sense disambiguation with WordNet (Fellbaum, 1998).

2. Classification of texts with ICD-10 codes, using many categories and extending the experiment with ontological knowledge. UMLS is particularly interesting in this particular case, as it has specific information on ICD-10 codes.

3. Distillation of knowledge from language models, capturing the knowledge inside the language model using the GNN, with a final architecture much smaller than an LLM. If enough knowledge of the language model can be captured in the GNN, an architecture that detaches the language model can perform the same classification task.

4. Explainable and traceable NLP models from well-defined graph architectures and their respective GNN. After training the model, inference can be made with new data, and the activation of the different components of the GNN can be seen to determine the prediction obtained as suggested by (Ying et al., 2019).

5. Optimise the proposed architecture to avoid over-fitting while training the classifier with datasets similar to the proposed one and coupling previously fine-tuned language models.

6. Consider alternative training methods for classification with a large number of labels, in this case, modifying the loss function according to the category to be classified ((Su et al., 2022), (Hüllermeier et al., 2020).

Acknowledgments

This research has been funded by the University of Alicante, the Spanish Ministry of Science and Innovation, the Generalitat Valenciana, and the European Regional Development Fund (ERDF) through the following funding: At the national level, the following projects were granted: Coolang (PID2021-122263OB-C22); CORTEX (PID2021-123956OB-I00); *CLEARTEXT* (TED2021-130707B-I00); and SOCIALTRUST (PDC2022-133146-C22), funded by MCIN/AEI/10.13039/501100011033 and, as appropriate, by ERDF A way of making Europe, by the European Union or by the European Union NextGenerationEU/PRTR. At regional level, the Generalitat Valenciana (Conselleria d'Educacio, Investigacio, Cultura i Esport), granted funding for NL4DISMIS (CIPROM/2021/21).

References

Agarwal, C., O. Queen, H. Lakkaraju, and M. Zitnik. 2023. Evaluating explainability for graph neural networks.

AlKhamissi, B., M. Li, A. Celikyilmaz, M. Diab, and M. Ghazvininejad. 2022. A review on language models as knowledge bases.

Bender, E. M., T. Gebru, A. McMillan-Major, and S. Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the*

2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Bodenreider, O. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

Chen, H., X. Liu, D. Yin, and J. Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *SIGKDD Explor. Newsl.*, 19(2):25–35, nov.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Elazar, Y., N. Kassner, S. Ravfogel, A. Ravichander, E. Hovy, H. Schütze, and Y. Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Fellbaum, C., editor. 1998. *WordNet: An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA.

Feng, Y., X. Chen, B. Y. Lin, P. Wang, J. Yan, and X. Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309, Online, November. Association for Computational Linguistics.

Gehman, S., S. Gururangan, M. Sap, Y. Choi, and N. A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In T. Cohn, Y. He, and Y. Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online, November. Association for Computational Linguistics.

Gu, Y., R. Tinn, H. Cheng, M. Lucas, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Trans. Comput. Healthcare*, 3(1), oct.

Gérardin, C., P. Wajsbürt, P. Vaillant, A. Bellamine, F. Carrat, and X. Tannier. 2022. Multilabel classification of medical concepts for patient clinical profile identification. *Artificial Intelligence in Medicine*, 128:102311.

He, H., H. Zhang, and D. Roth. 2022. Rethinking with retrieval: Faithful large language model inference.

Huang, N., Y. R. Deshpande, Y. Liu, H. Alberts, K. Cho, C. Vania, and I. Calixto. 2022. Endowing language models with multimodal knowledge graph representations.

Hüllermeier, E., M. Wever, E. L. Mencia, J. Fürnkranz, and M. Rapp. 2020. A flexible class of dependence-aware multi-label loss functions.

Jiang, X., Y. Shen, Y. Wang, X. Jin, and X. Cheng. 2020. Bakgrastec: A background knowledge graph based method for short text classification. In *2020 IEEE International Conference on Knowledge Graph (ICKG)*, pages 360–366, Los Alamitos, CA, USA, aug. IEEE Computer Society.

Kaur, J., S. Bhatia, M. Aggarwal, R. Bansal, and B. Krishnamurthy. 2022. LM-CORE: Language models with contextually relevant external knowledge. In M. Carpuat, M.-C. de Marneffe, and I. V. Meza Ruiz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 750–769, Seattle, United States, July. Association for Computational Linguistics.

Lee, E., C. Lee, and S. Ahn. 2022. Comparative study of multiclass text classification in research proposals using pretrained language models. *Applied Sciences*, 12(9).

Li, Y., D. Tarlow, M. Brockschmidt, and R. Zemel. 2017. Gated graph sequence neural networks.

Liu, F., E. Shareghi, Z. Meng, M. Basaldella, and N. Collier. 2021. Self-alignment pretraining for biomedical entity representations.

McCray, A. 1989. The umls semantic network.

Mikolov, T., K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space.

Mrkšić, N., D. Ó Séaghdha, B. Thomson, M. Gašić, L. M. Rojas-Barahona, P.-H. Su, D. Vandyke, T.-H. Wen, and S. Young. 2016. Counter-fitting word vectors to linguistic constraints. In K. Knight, A. Nenkova, and O. Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 142–148, San Diego, California, June. Association for Computational Linguistics.

Neumann, M., D. King, I. Beltagy, and W. Ammar. 2019. ScispaCy: Fast and robust models for biomedical natural language processing. In D. Demner-Fushman, K. B. Cohen, S. Ananiadou, and J. Tsujii, editors, *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy, August. Association for Computational Linguistics.

Peng, B., M. Galley, P. He, H. Cheng, Y. Xie, Y. Hu, Q. Huang, L. Liden, Z. Yu, W. Chen, and J. Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback.

Piad-Morffis, A., R. Muñoz, Y. Gutiérrez, Y. Almeida-Cruz, S. Estevez-Velarde, and A. Montoyo. 2019. A neural network component for knowledge-based semantic representations of text. In R. Mitkov and G. Angelova, editors, *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 904–911, Varna, Bulgaria, September. INCOMA Ltd.

Su, J., M. Zhu, A. Murtadha, S. Pan, B. Wen, and Y. Liu. 2022. Zlpr: A novel loss for multi-label classification.

Sun, J., C. Xu, L. Tang, S. Wang, C. Lin, Y. Gong, L. Ni, H.-Y. Shum, and J. Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *The Twelfth International Conference on Learning Representations*.

Sun, Y., S. Wang, S. Feng, S. Ding, C. Pang, J. Shang, J. Liu, X. Chen, Y. Zhao, Y. Lu, W. Liu, Z. Wu, W. Gong, J. Liang, Z. Shang, P. Sun, W. Liu, X. Ouyang, D. Yu, H. Tian, H. Wu, and H. Wang. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation.

Wang, L., W. Zhao, Z. Wei, and J. Liu. 2022. Simkgc: Simple contrastive knowledge graph completion with pre-trained language models.

Wang, X., Q. He, J. Liang, and Y. Xiao. 2023. Language models as knowledge embeddings.

Yasunaga, M., H. Ren, A. Bosselut, P. Liang, and J. Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In K. Toutanova, A. Rumshisky, L. Zettlemoyer, D. Hakkani-Tur, I. Beltagy, S. Bethard, R. Cotterell, T. Chakraborty, and Y. Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online, June. Association for Computational Linguistics.

Ying, Z., D. Bourgeois, J. You, M. Zitnik, and J. Leskovec. 2019. Gnnexplainer: Generating explanations for graph neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Yáñez Romero, F., A. Montoyo, R. Muñoz, Y. Gutiérrez, and A. Suárez Cueto. 2023-09. A review in knowledge extraction from knowledge bases.

Zhang, Z., X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu. 2019. ERNIE: Enhanced language representation with informative entities. In A. Korhonen, D. Traum, and L. Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy, July. Association for Computational Linguistics.

Zhou, J., G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81.