



**Università
degli Studi
di Palermo**

Dipartimento di Ingegneria
Direttore: prof. Livan Fratini



Dottorato in Information and Communication Technologies (ICT)
Ciclo XXXVII
Dipartimento di Ingegneria
Laboratorio di Interazione Uomo-Macchina

Towards the usage of Large Language Models in Information Retrieval and Question Answering task

Ph.D. Candidate:
Irene Siragusa

Ph.D. Coordinator:
Ch.ma Prof.ssa **Ilenia Tinnirello**

Supervisor:
Ch.mo Prof. **Roberto Pirrone**

Co-supervisor:
Ch.mo Prof. **Marco La Cascia**

CICLO XXXVII
ANNO CONSEGUIMENTO TITOLO 2025



Abstract

Natural Language Processing (NLP) field has deeply benefited of recent improvements in Deep Learning (DL), mainly involving transformers and the Attention Mechanism. Those technologies were crucial to assess outstanding performances in Information Retrieval (IR) and Question-Answer (QA) tasks.

Encoder-only transformers have obtained promising results in NLP traditional tasks, such as Relation Extraction (RE) and text classification. Language Models (LMs) if combined with suitable networks, have assessed their superior performance on these tasks, through the generation of contextual embeddings.

On the other hand, Large Language Models (LLMs) have shown impressive textual generation capabilities that lead to a redefinition of NLP tasks. QA with LLM benefits of Retrieval Augmented Generation (RAG) approach, through which models are capable of providing complete answers utilizing the given query and associated context.

Despite these promising results, a significant problem arises when dealing with both fine-tuning and evaluation of LLMs' generated text. Fine-tuning of such models become impractical due to their size, and novel strategies, such as Low-Rank Adaptation (LoRA) and prompt engineering, were developed to overcome the significant computational resources needed. On the other hand, the evaluation on LLMs is challenging for scenarios for which a golden answer is not available. The development of effective evaluation strategies, both involving automatic methods which involves a superior LLM as a judge, is still an open question in the scientific community.

Moreover, progress in LLM systems leads to the development of multimodal models, capable of jointly processing textual and visual inputs. On this basis, even more complex tasks can be performed, such as textual classification of the provided images, obtained by an analysis of the generated text.

This work focuses on the following principal Research Question (RQ): *how can NLP techniques, based on transformer models, be used to extract relevant semantic knowledge from unstructured text?*

At the beginning, the research objective was towards the usage of encoder-only models for classical NLP tasks such as RE and semantic classification. In this context, the main interest was in the development of simple architectures based on neural models in which the capabilities of LMs as feature extraction models,

were investigated and stressed. As for RE task, English-only models were involved in a classical data set for the target task, while the Italian language was the core of semantic classification in the context of the EVALITA competition for the detection of homotransfobic and hateful content on social networks.

As a result of the increasing popularity of generative models in the scientific community, the RQ was re-defined accordingly. In particular, an interest towards decoder-only models emerged towards the usage and performances of LLMs in complex tasks involving IR. In this context, tasks such as QA in a target domain with usage of the RAG approach were investigated. More in detail, considering the educational domain, LLMs’s capabilities were tested in both stand-alone and chatbot setups, using an *ad hoc* built data set. In these applications, selected LLM was queried with a custom instruction, the actual query, and the relevant context, thus a correct answer can be generated and the risks of hallucinations are reduced.

As multimodal models started been developed, QA and classification applications were developed in a multimodal setting for the medical domain. After properly arranging a multimodal data set of clinical cases, a RAG based pipeline was built for a multimodal setup. In this application, the retrieved context serves to build a suitable prompt to exploit in-context learning capabilities of the chosen LLM. The objective is to generate the correct prediction for the scanning modality and the body part shown in the given clinical case. In addition, a Knowledge Graph (KG) was automatically created using an LLM prompted accordingly to, essentially, perform a RE task in a generative manner. Obtained KG, serves as external knowledge source and its usage is explored in the context of a hybrid retrieval approach in the previous RAG-based architecture build. In this case, the model was asked to generate a disease hypothesis for the given clinical case.

To summarize the research objective presented in this work, the main problem of semantic IR in NLP field, was explored from two point of views, involving traditional NLP techniques and models, such as encoder-only LMs, and with the most recent developed LLMs. The use of such models, after the release of ChatGPT, cannot be explored or ignored, since it assessed a paradigm shift in the NLP field.

Contents

List of Figures	6
List of Tables	9
Acronyms	14
1 Introduction	18
2 All you need is Transformer	23
2.1 Transformers	23
2.1.1 Encoder-only and Masked Language Models	27
2.1.2 Decoder-only and Large Language Models	28
2.2 Retrieval Augmented Generation	34
3 Encoder-only models were all we had	37
3.1 Relation Extraction	38
3.1.1 SemEval 2010 Task 8	39
3.1.2 Related works	41
3.1.3 Methods	42
3.1.3.1 Encoding	42
3.1.3.2 Network architecture	43
3.1.4 Results	44
3.2 HaSpeeDe3	47
3.2.1 Methods	47
3.2.1.1 Pre-processing	47
3.2.1.2 Network architecture	49
3.2.2 Results	51
3.2.2.1 Error analysis	52
3.3 HODI	54
3.3.1 Methods	54
3.3.1.1 Pre-processing	55

3.3.1.2	Network architectures	57
3.3.2	Results	58
3.3.2.1	Error analysis	60
3.4	Discussion	61
4	Unipa-GPT framework	62
4.1	Related works	63
4.2	Unipa-corpus	66
4.2.1	Unipa-corpus for RAG	66
4.2.2	Unipa-corpus for fine-tuning	68
4.3	Unipa-GPT	71
4.3.1	Closed Unipa-GPT	71
4.3.1.1	Architecture	71
4.3.1.2	Results	73
4.3.2	Open Unipa-GPT	84
4.3.2.1	Evaluation data set	84
4.3.2.2	Partial Open Unipa-GPT	85
4.3.2.3	Complete Open Unipa-GPT	88
4.4	UniQA	97
4.4.1	Data set	97
4.4.2	Methods	102
4.4.2.1	Data split	102
4.4.2.2	Experimental setup	102
4.4.3	Results	106
4.5	Discussion	113
5	DR-LLM framework	116
5.1	Related works	117
5.2	MedPix 2.0	120
5.2.1	Data set	120
5.2.1.1	Data set extraction	121
5.2.1.2	Data set overview and tasks	125
5.3	CLIP over MedPix 2.0	127
5.4	DR-Minerva	129
5.4.1	Methods	129
5.4.1.1	Architecture	129
5.4.1.2	Experimental set-up	131
5.4.1.3	Metrics	132
5.4.2	Results	133
5.4.2.1	Modality classification task	134
5.4.2.2	Location classification task	135

5.4.2.3	Joint classification task	135
5.5	DR-Minerva with Knowledge Graph	137
5.5.1	Methods	137
5.5.1.1	Architecture	137
5.5.1.2	Knowledge Graph definition	138
5.5.2	Results	142
5.6	Discussion	147
6	Conclusions	150
6.1	Future works	153
7	Appendix A	155

List of Figures

2.1	Overview of the transformers architecture from [137]	25
2.2	Overview of the Scaled Dot Product Attention (a) and Multi Head Attention mechanism (b) from [137]	26
2.3	Overview of the RAG approach from [70]	34
2.4	Generic RAG architecture	35
3.1	Overview of a generic Relation Extraction pipeline	41
3.2	Example of encodings from a sample of SemEval 2010 Task 8 data set (a), namely textual (b) and concatenated (c) encoding	43
3.3	Overview of the proposed architecture for SemEval 2010 Task 8 for textual encoding (a) and concatenated encoding (ConEnc) (b)	44
3.4	Overview of the proposed architectures for HaSpeeDe3, (a) is made up of a module for embeddings extraction, a neural module and a classifier, while (b) adds an additional ReLU dense layer.	50
3.5	Overview of the proposed architecture for HODI (a) has a module for embeddings extraction, a neural module and a classifier while (b) adds an additional ReLU dense layer.	57
4.1	Overview of the details and course outline and documents for each corpora version.	67
4.2	The architecture of the RAG version of Closed Unipa-GPT.	71
4.3	The schema illustrates the architecture of the proposed system where GPT-3.5-turbo-fine-tuned is used after a fine-tuning process over Unipa-corpus.	74
4.4	The GUI of Unipa-GPT presented at the SHARPER Night 2023.	80
4.5	Charts illustrate the frequently asked topics to the model during the SHARPER Night 2023 (a), among the 31 feedback collected, an overview of the users is reported (b) and the rates to the application (c).	82

4.6	Charts illustrate the frequently asked topics to the model during the SHARPER Night 2024 (a), among the 31 feedback collected, an overview of the users is reported (b) and the rates to the application (c).	83
4.7	Overview of the Open Unipa-GPT architecture	89
4.8	Scatter plots of embedding spaces labeled as for typology from the Unipa-corpus	93
4.9	Scatter plots of embedding spaces labeled as for department from the Unipa-corpus	93
4.10	Schema of the implemented architecture for QA.	103
4.11	Scatter plots of 2D reduced embedding spaces, labeled for each language, from UniQA	106
4.12	Scatter plots of 2D reduced embedding spaces, labeled for each Department, from UniQA	107
5.1	Example from the MedPix [®] web site where the original screenshots have been re-arranged for visualizing purposes, and the colored labels have been added for clarity.	121
5.2	Example of a JSON document in MedPix 2.0 as for <i>description document</i>	122
5.3	Example of a JSON document in MedPix 2.0 as for <i>case-topic document</i>	123
5.4	An example of the interface for querying the database where the user asks for CT scans of female patients (a) and the provided output (b).	124
5.5	An example of the view of a selected clinical case retrieved by the query posed in Figure 5.4. The image, and the relative description are reported along with the information of the clinical case the image belongs to.	125
5.6	Overview of the principal features of CLIP: (1) Contrastive pre-training, (2) Creation of the classifier from text labels and (3) Zero-shot prediction. This schema is a re-adaptation of [106]	127
5.7	Overview of DR-Minerva architecture.	129
5.8	Multimodal sample from MedPix 2.0 re-arranged for DR-Minerva with the RAG documents and the inference query.	132
5.9	Overview of DR-Minerva architecture for the joint classification task.	137
5.10	Overview of the overall process for KG creation and inference.	138
5.11	Overview of the overall generated KG-10-N (a) and of its subgraph generated from the node “Pulmonary involvement” (b).	140
5.12	Overview of the subgraph generated from the node ‘Meningiomas’ in KG-10-N.	141

5.13	Overview of the generated architecture with RAG-based Flamingo and KG creation and inference.	141
5.14	Samples of end-to-end queries to the entire system leveraging KG-10-N and simple prompt.	146
5.15	Samples of end-to-end queries to the entire system leveraging KG-5-Y and complex prompt.	146

List of Tables

3.1	Statistics of SemEval 2010 Task 8 data set	40
3.2	Macro-averaged F1-Score for (9+1)-way classification over the test set of SemEval 2010 Task 8 with the proposed architectures.	45
3.3	Macro-averaged F1-Score for (9+1)-way classification over the test set of SemEval 2010 Task 8 with the SOTA models and the best proposed model.	46
3.4	Word distribution statistics over HaSpeeDe3 data set divided per label hateful (H) and not-hateful (NH).	48
3.5	Hashtags distribution statistics over HaSpeeDe3 data set divided per label hateful (H) and not-hateful (NH).	49
3.6	Emoji distribution statistics over HaSpeeDe3 data set divided per label hateful (H) and not-hateful (NH).	49
3.7	Overview of all the relevant results across the developed architectures on the golden labels from the HaSpeeDe3 test set. Starred results are the ones submitted to the competition, while the bold one is the highest score. Underlined results were obtained by removing stopwords from the data.	51
3.8	Macro F1 results over HaSpeeDe3 test set of the submitted models and the actual best one (starred). Both baseline result, ranking and expected ranking position are reported.	52
3.9	Word distribution statistics over the HODI data set divided per label.	55
3.10	Hashtags distribution statistics over HODI data set divided per label.	56
3.11	Emoji distribution statistics over HODI data set divided per label.	56
3.12	Overview of the best relevant results across the developed architectures on the golden labels from HODI test set. Italic values place below the baseline, while the bold one is the highest score. Underlined results were obtained by removing stopwords from the data.	59
3.13	Macro F1 results over HODI test set of the submitted models and their fixed versions (starred). Both baseline result, ranking and expected ranking position are reported.	59

3.14	Overview of the conducted experiments in Chapter 3.	61
4.1	Number of documents in each version of Unipa-corpus.	67
4.2	Generation prompts for Question-Answer (QA) pairs for the Education section	69
4.3	Number of documents in the Unipa-corpus-clear re-arranged for fine-tuning.	69
4.4	Overview of the instructions used in the Closed Unipa-GPT.	72
4.5	Evaluation results of the different runs for both Chat1 and Chat2 over Closed Unipa-GPT.	75
4.6	Answers from Chat2 generated with condensed instruction from base and FT GPT and the Unipa-corpus-emb for the retrieval. Second answer is truncated due to its length: in the missing text other cases of taxes reduction are illustrated.	77
4.7	Answers from Chat1 generated with custom instruction: the fine-tuned model contradicts itself.	78
4.8	Answers from Chat1 generated leveraging Unipa-corpus-clear as the retrieval corpus. Both the answers are correct but the one reported by the RAG-only model is more accurate.	78
4.9	Answers from Chat2 generated with base GPT-3.5-turbo and condensed instruction. Retrieval was performed, in the leftmost columns, with Unipa-corpus-emb and Unipa-corpus-full. The third model was queried only leveraging the fine-tuned version of GPT-3.5-turbo, with the same instruction used in training phase, i.e. without RAG.	79
4.10	The instructions used in the system during the SHARPER Night.	81
4.11	BLEU and ROUGE-L scores over baseline and open-source models as Generator LLM.	86
4.12	Context Relevancy scores over baseline and open-source models as Generator LLM. Higher values are marked in bold, underlined values marks the golden documents.	87
4.13	Faithfulness and Answer Correctness scores over baseline and open-source models as Generator LLM.	88
4.14	Context Relevancy scores over different Embedding LLMs. Higher values are marked in bold, underlined values marks the golden documents.	91
4.15	SC for each Embedding LLM with the two proposed clustering schemes, degree courses typology and affiliation to a particular Department.	92
4.16	BLEU, ROUGE-L, Faithfulness and Answer Correctness scores over the generated answers with baseline, open-source, and fine-tuned models	94

4.17	Faithfulness and Answer Correctness scores calculated with gemini-2.0-flash over the generated answers with baseline, open-source, and fine-tuned models	95
4.18	Overview of the document splits built from scraping the Unipa website.	98
4.19	List of the templates for QA pairs, which leverage the common textual header in each document couple in the English split.	98
4.20	List of the templates for QA pairs, which leverage the common textual header in each document couple in the Italian split	99
4.21	List of the templates for QA pairs in the Course info document, English split.	99
4.22	List of the templates for QA pairs in the <i>Course info document</i> , Italian split.	100
4.23	List of the templates for QA pairs in the Course outline document, English split.	100
4.24	List of the templates for QA pairs in the Course outline document, Italian split.	101
4.25	Distribution of the UniQA QA pairs in the two language splits. . .	101
4.26	List of the UniQA train-test splits grouped in Italian, English and Joint split	103
4.27	Clustering and retrieval performances of the three tested embedding LLM for UniQA.	107
4.28	BLEU and ROUGE score for the different LLMs used for evaluation of the UniQA QAs-JOINT-reduced split.	108
4.29	BLEU and ROUGE score for the different LLMs used for evaluation of the UniQA QAs-EN-reduced split.	108
4.30	BLEU and ROUGE score for the different LLMs used for evaluation of the UniQA QAs-IT-reduced split.	109
4.31	Example of a generated answer with fine-tuned models. R- refers to ROUGE Scores.	111
4.32	Overview of the conducted experiments in Chapter 4.	113
5.1	Summary of images and clinical cases for each split in MedPix 2.0. .	125
5.2	Summary of images divided per macro area of the body for each split in MedPix 2.0.	126
5.3	Overview of the two inference tasks reporting the used instruction prompt, considered labels, and accuracy score, after fine-tuning . .	128
5.4	The structure of the prompt is reported as well as the template of the corresponding expected answers	131

5.5	Overview of the used metrics in the context of DR-Minerva experiments expressed in terms of TP (True Positive), TN (True Negative), FP (False Positive) and FN (False Negative), and considering C as number of classes.	133
5.6	Experimental results for MedPix 2.0 test set in modality classification task. The starred line represent the inference in a zero-shot evaluation mode.	134
5.7	Experimental results for MedPix 2.0 test set in location evaluation setting. The starred line represent the inference in a zero-shot evaluation mode.	135
5.8	Experimental results for MedPix 2.0 test set in joint classification task. The starred line represent the inference in a zero-shot evaluation mode.	136
5.9	The structure of the prompt is reported as well as the template of the corresponding expected answers	138
5.10	Overview of the hyperparameter setups used to generate the different versions of the KG.	139
5.11	Summary of BERT metrics calculated on all KG setups for each of the proposed instruction prompts. Bold values indicate the best score for each KG and body part respectively	144
5.12	Summary of the overall calculated metrics on all KG setups for each of the proposed instruction prompts and a baseline reference. Bold values indicate the best score for each metric	145
5.13	Overview of the conducted experiments in Chapter 5.	149
7.1	English version of Table 4.4.	155
7.2	English version of Table 4.6.	156
7.3	English version of Table 4.7.	156
7.4	English version of Table 4.8	156
7.5	English version of Table 4.9.	157
7.6	English version of Table 4.10.	157
7.7	List of the six QA pairs used for evaluation purposes	158
7.8	English version of Table 7.7.	159

List of acronyms

AI	Artificial Intelligence
ACC-J	Accuracy Joint
ACC-L	Accuracy Location
ACC-M	Accuracy Modality
BiLSTM	Bidirectional LSTM
BCE	Binary Cross Entropy
CNN	Convolutional Neural Network
ConEnc	Concatenated Encoding
CoT	Chain of Thought
CQA	Closed Question Answering
CRFs	Conditional Random Fields
CT	Computed Tomography
DB	Database
DL	Deep Learning
DNN	Deep Neural Network
DR-Minerva	Minerva Diagnostic Retriever
EOS	End Of Sequence
FAISS	Facebook AI Similarity Search
F1-J	F1-score Joint

F1-L	F1-score Location
F1-M	F1-score Modality
GCN	Graph Convolutional Network
GNN	Graph Neural Network
GPT	Generative Pre-training Transformer
HaSpeeDe	Hate Speech Detection
HODI	Homotransphobia Detection in Italian
IR	Information Retrieval
HMM	Hidden Markov Model
KB	Knowledge Base
KG	Knowledge Graph
LLM	Large Language Model
LM	Language Model
LoRA	Low-Rank Adaptation
QLoRA	Quantized Low-Rank Adaptation
LSTM	Long Short Term Memory
MLM	Masking Language Modeling
MR	Magnetic Resonance
MTEB	Massive Text Embedding Benchmark
NER	Named Entity Recognition
NLP	Natural Language Processing
NSP	Next Sentence Prediction
OOV	Out Of Vocabulary
OQA	Open Question Answering

PEFT	Parameter-Efficient Fine-Tuning
POS	Part Of Speech
RLHF	Reinforcement Learning with Human Feedback
PREC-J	Precision Joint
PREC-L	Precision Location
PREC-M	Precision Modality
PTLM	Pre-Trained Language Model
QA	Question-Answer
RAG	Retrieval Augmented Generation
RE	Relation Extraction
REC-J	Recall Joint
REC-L	Recall Location
REC-M	Recall Modality
RQ	Research Question
RUS	Reproductive and Urinary System
SaM	Spine and Muscles
SC	Silhouette Coefficient
SOTA	State-Of-The-Art
TexEnc	Textual Encoding
VLM	Visual Language Model
VQA	Visual Question Answering
Unipa	University of Palermo

Chapter 1

Introduction

Recent innovations in Artificial Intelligence (AI) have mainly involved Natural Language Processing (NLP) that gained widespread diffusion among researchers and non-academics. This phenomenon was particularly emphasized since ChatGPT models [14, 97] were released and general interest in AI systems highly increased. Analogously as Convolutional Neural Network (CNN) [68] and AlexNet [62] are considered a crucial turning point in Deep Learning (DL) application in Artificial Vision in 2012, the Attention Mechanism in transformer architecture, proposed in 2017 in “Attention is all you need” [137] also marks a before and after in NLP field.

After this publication, the potentialities of transformer-based systems, such as encoder-only models like BERT [27], leveraging the novel Masking Language Modeling (MLM) strategy, established a new State-Of-The-Art (SOTA) methodology in NLP research.

Although the idea of training neural systems for embeddings generation for NLP tasks was not recent, developed models used static embeddings, which showed significant limitation, mainly related with dealing with Out Of Vocabulary (OOV) words. The introduction of the attention mechanism, along with the encoder-based models, allowed the training and development of newer Language Models (LMs) using the MLM strategy, and obtaining the so-called contextual embeddings. On this basis, an embedding to a given word is not static, but it is calculated on the basis of its surrounding words. In addition, the combination of this methodology, combined with tokenization strategies based on subwords, allows the extraction of OOV words. Given the word *hyperparameters*, which is a OOV for a given LM, e.g. BERT, it can be split into subwords such as *hyper + para + meter + s*, for which a comprehensive embedding representation can be derived, for example as mean of the embedding of the subwords.

On the other hand, the development of decoder-only models assesses a new standard in textual generation modeling that achieves impressive performances combining the autoregressive training strategy with Reinforcement Learning with Human Feedback (RLHF) [99]. Despite this success and a growing interest in these models from the scientific community, two practical issues arise: fine-tuning and evaluation. The former is mainly related to the size of those models that require larger GPUs and data to train, fine-tune, and inference them. The latter emerges when dealing with tasks for which an evaluation process is needed but is impractical. The unavailability of golden labels and a large test set would require a manual evaluation that is not reasonable or a newer evaluation paradigm.

In addition, generative models allowed for a re-definition of traditional discriminative task, e.g. text classification. In this context, instead of designing a suitable neural network, the idea of simply asking to ChatGPT became an option. In this scenario, a newer paradigm change can be observed towards a preference for generative models that can be properly trained via Parameter-Efficient Fine-Tuning (PEFT) techniques [74] or through prompt engineering, using an in-context learning approach enriched via example in a few-shot learning design [14]. Essentially, the idea is to provide to the generative model a textual instruction from which contextually understand the task to perform (in-context learning), and some examples that show how the model is expected to reply as for the answer construction from both a morphological and semantic perspective (few-shot learning).

This doctoral dissertation explores textual and multimodal methodologies for Information Retrieval (IR) in target domains using both encoder- and decoder-only models. To fully understand the context of development of the presented works, a temporal background is required. During the first PhD year, the principal Research Question (RQ) was formulated along with the study of methodologies and SOTA techniques in NLP field and first experimental setups were defined. At the end of 2022, which coincides with the start of the second PhD year, ChatGPT was made available worldwide [14].

Inevitably, generative decoder-only models, such as GPT [14, 97] and Llama [133, 82] started to gain popularity in the scientific community. Those models, also referred to as Large Language Model (LLM), progressively assess a new generative paradigm for traditional NLP tasks, that originally were treated with encoder-only models [48].

This enormous change throughout 2023, deeply and inevitably influenced the research work: approaches with encoder-only models were discarded in favor to decoder-only models. This turning point was further stressed as newer research fields, involving textual and multimodal QA, were explored. In this context, the

focus was on building chatbot applications for the educational and medical domain with Retrieval Augmented Generation (RAG) approach.

The objective of this work can be found in the following principal RQ: *how can NLP techniques, based on transformer models, be used to extract relevant semantic knowledge from unstructured text?*

This RQ was split into simpler and smaller tasks, which are the core of each chapter in this thesis and are reported in the following.

RQ 1 *How to perform an English Relation Extraction (RE) task and a semantic classification task in Italian in a resource-limited environment and utilizing encoder-only models?*

Techniques involving encoder-only models and Deep Neural Networks (DNNs), are explored in the context of the English-based challenge SemEval 2010 Task 8 [45] in Section 3.1. In addition to the definition of the task, two diverse encoding methodologies were proposed.

While, in the context of EVALITA 2023 campaign for the evaluation of NLP tools for the Italian language, two challenges involving the classification of Italian tweets have been chosen, namely HaSpeeDe3 [65] and HODI [95]. The tasks has been approached following a *minimalist* strategy to detect the presence of hateful and homotransfobic content in Italian tweets [127, 122]. Italian LMs were used as embedding generator model and a DNN, such as LSTM, BiLSTM, transformers and dense layers, was implemented for further processing the obtained embeddings to obtain the final classification. Related semantic analysis over the two considered data set and overview of developed models, are described in Sections 3.2 and 3.3. While, in the context of EVALITA 2023 campaign for the evaluation of NLP tools for the Italian language, the two Italian tweets data sets and their classification challenge, namely HaSpeeDe3 [65] and HODI [95], have been chosen a *minimalist* approach was proposed for detection of hateful and homotransfobic content in Italian tweets [127, 122]. Italian LMs were used as embedding generator model and a DNN, such as LSTM, BiLSTM, transformers and dense layers, was implemented for further processing the obtained embeddings to obtain the final classification. Related semantic analysis over the two considered data set and overview of developed models, are described in Sections 3.2 and 3.3. While, in the context of EVALITA 2023 campaign for the evaluation of NLP tools for the Italian language, the two Italian tweets data sets and their classification challenge , namely HaSpeeDe3 [65] and HODI [95], have been chosen a *minimalist* approach was proposed for detection of hateful

and homotransphobic content in Italian tweets [127, 122]. Italian LMs were used as embedding generator model and a DNN, such as LSTM, BiLSTM, transformers and dense layers, was implemented for further processing the obtained embeddings to obtain the final classification. Related semantic analysis over the two considered data set and overview of developed models, are described in Sections 3.2 and 3.3.

RQ 2 *How can we use a decoder-only model to develop a chatbot and a QA application with domain-based IR for educational domain?*

Unipa-GPT framework [123, 125, 124] represents the work in the context of educational domain in which both closed- and open-source models, where tested in a chatbot setup with the *ad hoc* build Unipa-corpus of documents inherent to the University of Palermo (Section 4.2). Leveraging the RAG approach, relevant document to the given query of a user, are selected and used as a context to generate the correct answer. In addition to methodologies to build the corpus and to develop the chatbot, and collected feedback from people who used the chatbot during the SHARPER Night in 2023 and 2024, are reported in Section 4.3.

The usage of decoder-only models in a QA setup was more explored thanks to the UniQA data set [126]. This data set, build from the obtained English and Italian documents of the Unipa-corpus, represents a set of Italian and English QA data set, which allows a comprehensive evaluation of the generated answer, thanks to the availability of golden answers. Building process of the data set and its evaluation via a RAG-based pipeline is described in Section 4.4.

RQ 3 *How can we use a multimodal model with domain-based IR for the biomedical domain, and enhanced a decoder-only model to provide a diagnostic suggestion?*

Multimodal models are the core of the DR-LLM framework, in which Med-Pix 2.0 [119], a data set of clinical images and reports, was built, and used to perform a classification task to determine the typology of the given medical image and the body part shown (Section 5.2). To enhance model's capabilities, a domain-based IR was performed, to enrich the prompt given in input to the model itself in a in-context learning setup [121]. Building process of the data set, task definition and model's developing and testing is illustrated in Section 5.4.

Leveraging the previous obtained results, a strategy to enhance the generated output of the decoder-only model was developed in a RAG setup in which an automatically built Knowledge Graph (KG) was used as external

knowledge source [120]. Building process of KG relies on a RE technique in which a decoder-only model is query accordingly to extract relevant triplets. Developing process of this pipeline is reported in Section 5.5.

As stated before, the enormous change in 2023 led to a change towards decoder-only models, and the principal RQ was adapted accordingly: the focus towards IR was re-arranged towards the educational and biomedical domain, where the data set, document creation, model selection, prompt design and evaluation techniques were the core of Unipa-GPT and DR-LLM frameworks. In addition, the building process of KG, which can be used as an external source in a RAG-based architecture, can be seen as a semantic application of RE techniques performed in a generative manner.

In this perspective, traditional NLP tasks serve as small building blocks for performing more complex tasks, for which multiple agents are required to obtain the desired result, as in Section 5.5.

In addition to the previously reported RQs, an overview of the main theoretical topics, such as the principal characteristic of the diverse LMs and LLMs used in this work, and an analysis on the RAG approach are described in Chapter 2, while concluding remarks are given in Chapter 6.

Chapter 2

All you need is Transformer

In this chapter a State-Of-The-Art (SOTA) overview of the principal theoretical topics discussed in subsequent chapters is reported. More in detail, the main characteristics of the Transformers model are presented in Section 2.1, together with a comprehensive overview of the characteristics of the encoder-only and decoder-only models used in this work. In addition, characteristics and benefit of the Retrieval Augmented Generation approach used with Large Language Models, are illustrated in Section 2.2.

2.1 Transformers

Computational Linguistics deeply leverages and benefit of Deep Learning (DL) techniques to better perform in traditional Natural Language Processing (NLP) tasks such as Part Of Speech (POS) tagging or Named Entity Recognition (NER) with Hidden Markov Model (HMM) [22] or Conditional Random Fields (CRFs) [64]. Moreover, the idea of embeddings, which are dense numerical vectors, used to represent words in a more meaningful way than one-hot encoding or bag of words, starts to spread with the introduction of Word2Vec [90] and GloVe [90]. Those models, trained on a suitable corpus, create a static embedding for a given word, thus allowing the usage of more sophisticated Deep Neural Network (DNN) architectures and obtaining an overall increase in performances of such developed models. In addition, those models, moreover, showed semantic properties in the form of Euclidean distance: for example, the result of the expression $\overrightarrow{king} + \overrightarrow{man} + \overrightarrow{woman}$ is a vector close to $\overrightarrow{queen}$ [90]. A well-trained embedding model projects to the obtained numerical vectors, some linguistic semantic properties, locating synonyms and similar words as for their significance, closer to each other and far from their opposite.

Despite this powerful mathematical relation with semantics, these models are not able to extract embeddings from an unknown word. The problem related with Out Of Vocabulary (OOV) words, which are words on which the model was not trained on, obliges a fine-tuning process to provide the missing knowledge, but still remains prone to mistakes when dealing with more unknown words at inference time.

Models, such as fastText [40], are a step towards OOV words and sparsity problems from previous models: they take advantage on the subwords model, through which an OOV word can be decomposed and processed correctly.

Notwithstanding the general promising results, another issue still remains: static word embeddings, despite a suitable training procedure, provide the same numerical embedding to a given word, without leveraging the context contribution of the specific phrase where target word is extracted from. Moreover, in the case of a word with multiple meanings or semantic and metaphorical nuances, those changes in its sense cannot be properly caught with static embeddings.

On this basis, the idea of contextual embedding seems to be interesting, since nuanced meaning can be provided to a single word according to both its specific meaning in a target context and the context itself. A valuable method to do this is with Transformers [137] that combines an encoder and a decoder to process textual embedding sequences combined with their positional encoding, as in Figure 2.1.

In addition to the proposed architecture, the Scaled Dot Product Attention, also called “Attention Mechanism”, was implemented (Figure 2.2.a) and allows the model to generate embedding for a given word, that are aware of the context; i.e. embedding for w_i depend on both previous $w_{i-1}, w_{i-2}, \dots$ and subsequent $w_{i+1}, w_{i+2}, \dots$ words. More specifically, given two input embeddings $\mathbf{x}_i$ and $\mathbf{x}_j$, three different representations are calculated via projection matrices, thus obtaining its query ($\mathbf{q}_i$), key ($\mathbf{k}_i$) and value ($\mathbf{v}_i$) representations. The representations obtained are stored in the corresponding matrices Q , K , and V and then combined to obtain a score representing the mutual relevance of a given $\mathbf{x}_i$ with respect to all others $\mathbf{x}_j$ in the given sequence

$$Attention(Q, K, V) = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

where d_k is the dimension of the key vector. The proposed attention mechanism is extended to the entire sequence in the form of Multi Head Attention (Figure 2.2.b), and transformer-based models, leverages on a stack of transformers to obtain the final desired representation.

Transformer models can use the proposed attention mechanism in three ways,

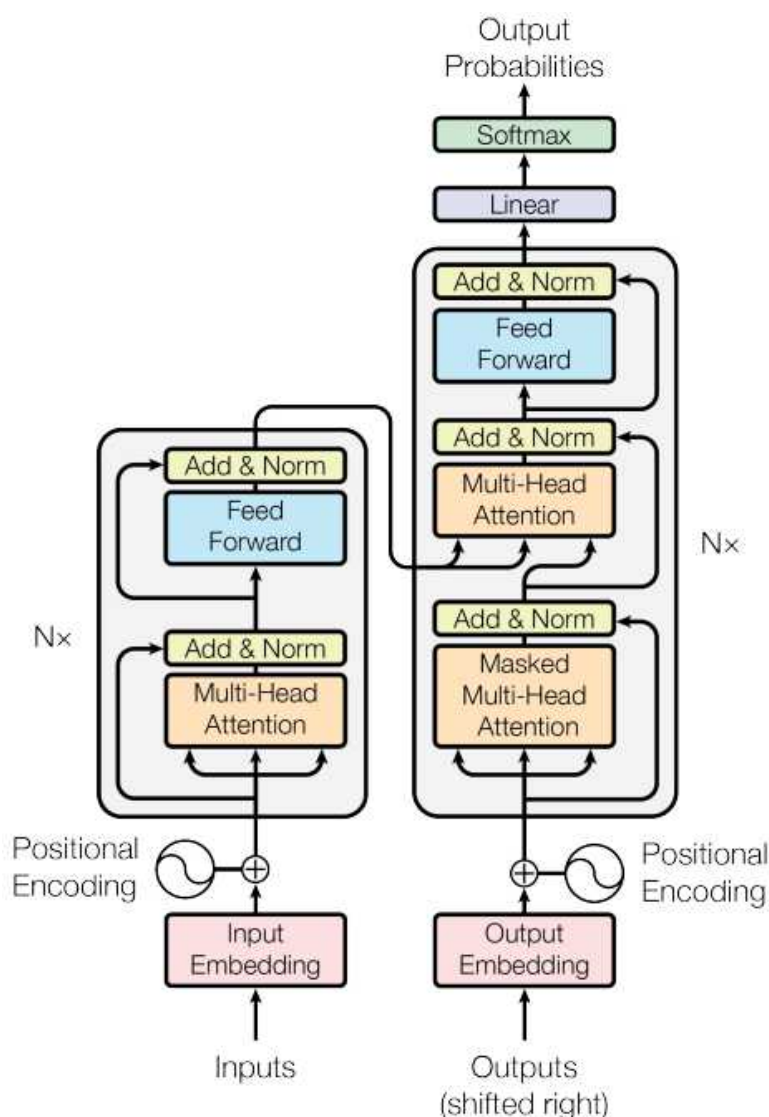


Figure 2.1: Overview of the transformers architecture from [137]

thus defining encoder-only [27], decoder-only [107], and encoder-decoder models [137, 109].

Encoder-only models employ a Bidirectional self-attention mechanism in which the attention score for a given word is calculated on both left and right contexts.

In contrast, decoder-only models use a causal self-attention mechanism in which only the left-context is considered to calculate attention for a given word [27, 58]¹. In this context, encoder-only models, with their intrinsic bi-directionality

¹p. 322

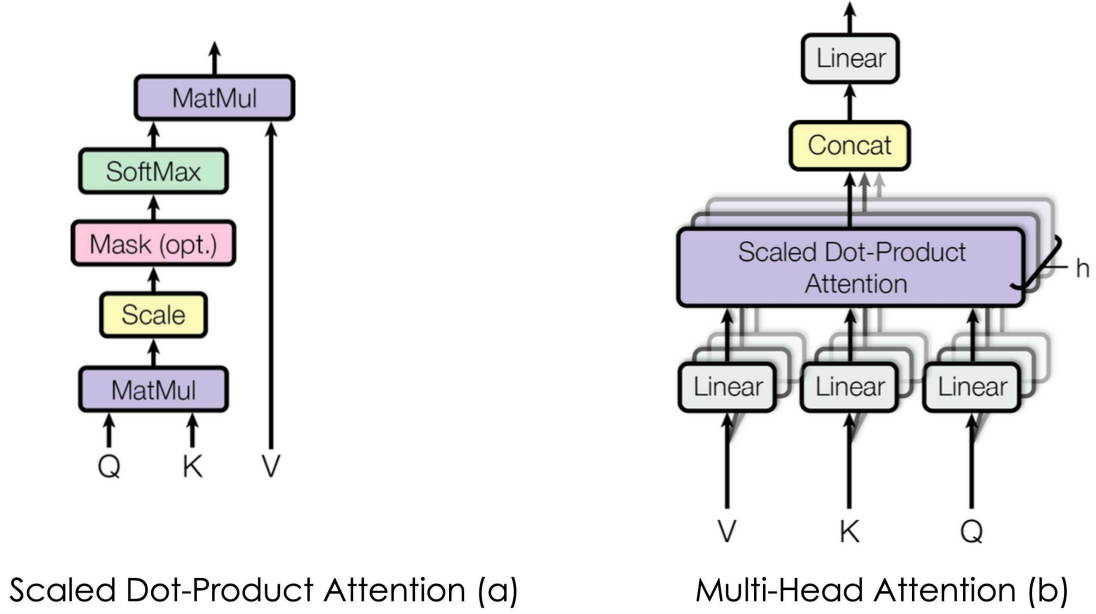


Figure 2.2: Overview of the Scaled Dot Product Attention (a) and Multi Head Attention mechanism (b) from [137]

approach became suitable for contextual embedding representation, while decoder-only models, for generative application, in which newer tokens are generated from previous ones.

Encoder-decoder models combine aspects of the two aforementioned architecture and they add a cross-attention layer in the decoder, in which the self-attention layer is applied between query representation from the decoder itself and key and value representation from the last encoder layer, in which only the left-context is taken into account. Encoder-decoder models and cross-attention strategy are useful when dealing with sequence-to-sequence tasks such as machine translation, which highly benefit these systems.

In this chapter and in the subsequent ones, encoder- and decoder-only models will be analyzed and considered for the experimental phase. More in detail, encoder-only models are used as for embedding generation for Relation Extraction and semantic classification tasks in Chapter 3, while decoder-only models are used for generative purposes in educational (Chapter 4) and medical (Chapter 5) domains.

2.1.1 Encoder-only and Masked Language Models

Encoders-only models, also referred to as discriminative, represent the first example of Pre-Trained Language Model (PTLM) or Language Model (LM) obtained with transformer architecture combined with the attention mechanism. The first developed model in this context is BERT [27], which was trained with two training strategies, namely Masking Language Modeling (MLM) and Next Sentence Prediction (NSP).

As for MLM, starting from an unannotated text, 15% of random tokens are sampled from each training sequence and three different learning tasks are designed. In the first case, the sampled token is replaced with the [MASK] token, in the second one is replaced with another token randomly sampled from the vocabulary based on token unigram probabilities, and, in the third case, sampled tokens are left unchanged. Despite the learning task, LM has the objective to predict the original input for each masked token, as in the following example:

[CLS] My dog [MASK] cute and likes to play with his ball. [SEP]

model is forced to fill the [MASK] token with *is*, which is the correct one. Moreover, sequences are enriched with two special tokens, namely [CLS] and [SEP]. [CLS], which is located at the beginning of the sequence, is trained to provide a comprehensive embedding of the entire input sequence, which serves as input to a classifier head for subsequent downstream tasks. [SEP] token, instead, is located at the end of the sequence serving as End Of Sequence (EOS) token, and as a separation token between multiple sequences if they are fed into the model.

More specifically, [CLS] and [SEP] tokens play a crucial role in NSP training strategy, in which the model learns text-pairs representation- Here the objective is to determine if the first sentence is followed by the second one: [SEP] serves as placeholder to separate the two sentences, and [CLS] is a unique representation of the processed input. After a pre-training process, BERT model can be further fine-tuned for downstream tasks such as text classification or Question-Answer (QA).

After BERT, researchers developed newer variants, exploring the adaptability of PTLM for domain-specific applications, such as BioBERT, which is trained on biomedical corpora [69], or different training strategy.

RoBERTa [81], is a robust version of BERT, obtained by enhancing and stressing the pre-training approach proposed in [27]. SpanBERT [56], designed to better represent and predict text spans, differs greatly from BERT for masking and training strategies, which is span level instead of token level.

Although the previously reported models are English-oriented, the research community is also interested in exploring multilingual models, thus enlarging the training set with multilingual data, models such as multilingual BERT (mBERT) [27] and XLM-RoBERTa [23] were proposed.

In addition to these models, models specifically oriented to a target language were also investigated and developed. In this context, works such as BERT-it [118] and ALBERTo [104], represent an adaptation of BERT-like approaches towards Italian. Both models built a custom vocabulary and tokenizer over which the developed models were trained on Italian data from Wikipedia and Twitter, respectively.

Regardless of the specific training strategy to develop an encoder-only model, all the aforementioned ones shares the same bi-directional paradigm at inference time. Bi-directionality, which is connected to the self-attention mechanism, assures that attention weights associated to a given word depend on both previous and subsequent words. Following this overview, encoder-only models can be seen as bi-directional models for which the input sequence can be read both left-right and right-left, making them suitable for encoding and solving NLP tasks such as text or sequence classification.

The models reported so far are the building blocks used for the neural architectures proposed in Relation Extraction tasks (Section 3.1) and for semantic classification in the context of the EVALITA competition for the detection of hateful content (Section 3.2) and homotransphobia (Section 3.3) in social networks. More in detail, BERT, SpanBERT, RoBERTa, and XLM-RoBERTa will be used for Relation Extraction with SemEval 2010 Task 8 data set [45], while XLM-RoBERTa, fastText, ALBERTo, mBERT, and BERT-it, in the Italian-based tasks proposed in HaSpeeDe3 [65] and HODI [95] data sets.

2.1.2 Decoder-only and Large Language Models

Decoder-only models are generative language models capable of predicting and generating newer words, from the given input, usually referred to as Large Language Model (LLM). This is generally formalized with the chain rule as follows.

$$P(w_{1:n}) = \prod_{i=1}^n P(w_i | w_{1:i-1})$$

But when dealing with generative models, more specifically at inference time, the so-called autoregressive or casual generation technique is used. Within this gener-

ation strategy (Algorithm 1) the word generated at each time step is conditioned on the word selected by the model itself from the previous step [58]².

Algorithm 1 Overview of the autoregressive algorithm

```

 $i \leftarrow 1$ 
 $w_i \sim P(w)$ 
while  $w_i \neq \text{EOS}$  do
   $i \leftarrow i + 1$ 
   $w_i \sim P(w_i | w_{<i})$ 
end while
  
```

As stated by the provided algorithm, words are generated according to the conditional probability of the previous generated ones, till the EOS token is generated, thus stating that the generation process is completed.

Calculations of conditional probabilities are also referred to as the decoding strategy and may involve various techniques to determine the next generated word. The decoding strategy may be classified as deterministic or stochastic: the former can be reproduced, while the latter adds randomness and allows for more creative answers. In the context of this work, creative will refer to an answer generated from a LLM in a nondeterministic manner, with the result being unpredictable and with a higher degree of freedom than for the generating parameters.

The simplest decoding strategy is greedy decoding in which at each step the generated word is the most probable one. Beam-search can be derived as an enriched version of greedy decoding. In this case, where several hypotheses are explored till the one with the overall highest probability for the entire sequence is selected to be the generated answer. This process results in a more sophisticated one, since high-probability sequences, starting with tokens with lower probability, can be selected.

Among the stochastic decoding strategies, sampling adds a random element that may result in the generation of much more interesting outputs. For example, in top- k sampling, the k most probable words are selected, their generation probabilities are normalized, and the next word is randomly sampled from the created distribution. In doing this, the generative strategy is extended to a set of the most probable words, allowing diverse high-quality text generation.

Other relevant generation algorithms that are worth mentioning are the nucleus or top- p sampling and the temperature sampling. The former, instead of truncate the first most probable words, keep the first words that cover the top p percent

²pp. 167-168, 207-208

of the probability mass. The latter, on the contrary, do not truncate the distribution but reshape it according to a temperature value t that modifies the logit values before applying the softmax through which the final prediction is obtained. As in thermodynamics, a higher temperature level corresponds to more flexibility in generation, that is, resulting in a more creative model, while low temperature values implies a more standardized and predictive answer [58]³.

Generative LLMs can be divided into foundational and instruct models: the former, also referred to as base, mainly focuses on generic language generation, the latter on generating a proper text according to a given instruction. As for foundational models, they are trained according to a self-supervised strategy combined with teacher forcing. In self-supervision, the training process is performed without annotated text since the model learns how to predict the next word. During training phase, if the model gets the wrong prediction, weights of the model are updated accordingly to the error back-propagated with loss, but the training continues via providing the model the correct sequence, i.e. the error is ignored for the subsequent next-word prediction. Instruct models, on the contrary, require an annotated data set to be trained on, comprehensive of a structured input called prompt.

In the context of this work, a prompt refers to the input provided to an LLM made up of an instruction, which describes the role assigned to the LLM and its desired behavior, and an actual input that can be considered as the query or input of the user. In addition to these elements, a prompt can be enriched by a context or by some example in a few-shot setting. Both components bring useful information to the model on which it can rely for generating an answer: the former focuses on providing to the model some target knowledge to address the generative process from a semantic perspective. The latter, on the contrary, deeply conditions the generation process from a morphological point of view, since some input-output pairs are provided to the model along with the prompt, thus forcing the model to answer to the actual input by replicating the morphological aspects of the given examples. Assuming the following prompt:

Classify the following lyrics, specifying whether they are written in Italian or English.

“Is this the real life? Is this just fantasy?” - English

“Parlami, come il vento fra gli alberi” - Italian

“I’m gonna pick up the pieces and build a LEGO house”

The model is expected to provide the correct classification label *English*, in the

³pp. 207-210

desired form, which is - *English* and not as *this text is written in English*, imitating the style of the given examples.

Decoder-only models gained worldwide popularity with the release of ChatGPT [14], which gained much interest from both the researcher and the general public. It can be seen as the first time that an LLM went out from the scientific community and reached a general consensus among those who used it. The model behind ChatGPT is an instruct model belonging to Generative Pre-training Transformer (GPT) model family, that was trained with a Reinforcement Learning with Human Feedback (RLHF) [99] strategy in which model is not only forced to follow the given instruction, but also to provide answers that align with human preferences. The alignment with human preferences was crucial for the popularity of ChatGPT and the increasing trust and reliability on using an Artificial Intelligence (AI) system.

Despite those strength points, when dealing with LLMs newer problems appeared: the term Large in LLM, emphasizes the size of these models that reach the parameters of *only* 4B for small models, ranges from 7B to 9B for medium-small models, to reach a consistent size of 70B or 175B as for GPT 3. Connected to the size of models, troubles related to the training data and the computational resources needed to train those models arise. This results in a small set of developers that propose their large and powerful LLMs such as, among others, the GPT family from OpenAI [107, 108, 14, 97], Llama base and chat- and instruct-fine-tuned models from Meta [133, 82], PaLM and Gemini from Google [20, 36], Claude from Antropic [3] and Mistral from Mistral AI [53].

Excluding Meta and Mistral AI, these models can be considered as closed models, since they can be inferred and trained only via paid API requests. In contrast, Llama and Mistral models public released their models, thus allowing a wide investigation of the adaptability of these models to various domains, such as generic instruction fine-tuning as Alpaca [132] or chat-oriented as Koala [38] or Baize [145] from Llama 2.

Moreover, leveraging all the released versions of Llama models, diverse adaptations to the Italian language were proposed such as Llamantino from Llama 2 base, Llamantino-chat [8] and Fauno [5] from Llama 2 chat, via fine-tuning with Italian data sets. Camoscio [115] was obtained via instruct fine-tuning from Llama 2 using the same approach as in [132] while Anita [105] is an Italian adaptation of Llama 3 instruct. In contrast, some attempts to train from scratch, using an existing architecture, have been made in the context of Minerva models [98]. This family of LLM uses a Mistral-based architecture and has been trained for the Italian language from scratch with an open data set divided equally into English and Italian data.

To effectively manage the fine-tuning of such large models, the scientific community developed various effective fine-tuning strategies, referred to as Parameter-Efficient Fine-Tuning (PEFT), which involves freezing, quantization and Low-Rank Adaptation (LoRA) [74].

Freezing is the process through which weights of selected layers of the model are frozen and thus are not involved during fine-tuning, allowing a lighter process from both memory and time. Analogously, quantization [51] reduces the memory involved in the training process by reducing the precision of the weights in the net from a 32-bit representation to a 16-, 8- or 4-bit representation.

Quantization, in this context, can allow for a full fine-tuning by reducing the memory occupation, and without need of freezing is needed. Moreover, this technique can also be used at inference time, thus allowing inference of 7B LLM in a low-resource environment such as a 16GB GPU.

In addition, LoRA [49] is one of the most used PEFT technique in which weights of selected layers of the architecture are kept frozen while the corresponding low-weight adapters are trained. More in detail, a pre-trained layer can be represented as a $d \times k$ matrix, and a rank-based decomposition can be performed through two matrices, with dimensions of $d \times r$ and $r \times k$ being created. On this basis, the number of trainable parameters became $(d \times r + r \times k) \ll (r \times k)$ if r is sufficiently small.

In fact, it is common to combine these techniques to better fit computational constraints, such as in Quantized Low-Rank Adaptation (QLoRA) [26] in which quantization is applied along with LoRA during training.

In conclusion, also prompting strategies, such as in-context learning [14], can be referred to as another PEFT technique [74]. Prompt engineering consists of determining the best possible prompt through which ask an LLM to execute a certain task, without any training process. Prompts can be enriched with some example, thus implementing an in-context learning approach [14, 28], in which the ability of LLM to learn from a provided instruction and a few examples, as in a few-shot scenario. Other strategies involving the usage of Chain of Thought (CoT) [143], that consists is forcing the model to execute a series of intermediate reasoning steps, thus improving its abilities for tasks in which a complex reasoning is required. Different inferences with different prompts lead to different generated answers (Autoregressive algorithm 1), thus even the slightest change in the prompt produces enormous changes in the final generated answer. In addition, an important role is played by the decoding strategies involved at inference time, along with the selected hyperparameters according to the specific generative task and type of desired output.

Generative models reported so far are the building blocks for the Unipa-GPT framework in the educational domain (Chapter 4) and for the DR-LLM framework in the medical domain (Chapter 5). More in detail, multilingual and Italian-based LLMs, such as OpenAI GPT models, Llama models, LLaMantino, Fauno, Camoscio, Anita and Minerva, were used in the context of Unipa-GPT. Minerva and Llama models, which natively support the English language, were involved in the development of the models in the DR-LLM framework.

2.2 Retrieval Augmented Generation

Generative capabilities of LLMs are quite impressive, but inaccurate or erroneous answers can be generated if models are asked to provide answers related to specific topics. This phenomenon is referred to as hallucination [52, 152] and finding effective methods to overcome or mitigate this undesired behavior is a new Research Question (RQ) in NLP field. A generative LLM essentially outputs the most probable word given its predecessors, and inevitably, higher probabilities are associated to pattern familiar for the model, i.e. to something highly correlated with its training set and training strategy. Generally speaking, generated output from a LLM often results as a plausible statement, obtained with the attempt to emulate its internal knowledge, acquired during its training phase, that does not necessarily correspond to a truthful output. In fact, generalization capabilities of LLMs, allow them to generate answers that seem to be correct from a grammatical perspective, and sometimes quite convincing and persuasive, so that a human can consider the provided answer as truthfully if the golden answer is unknown.

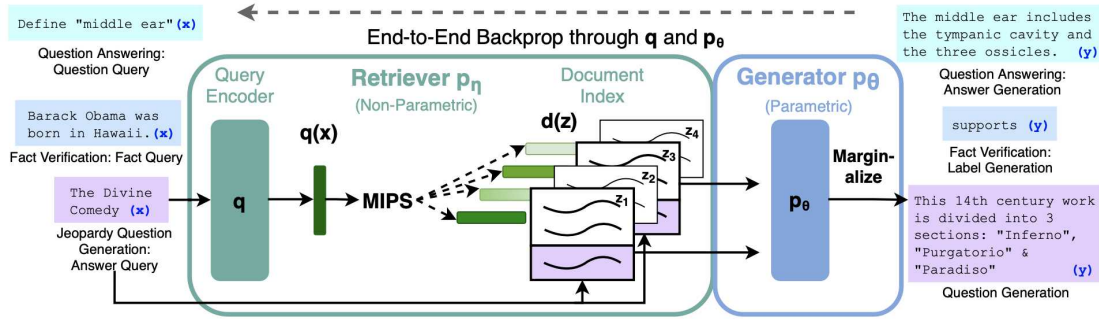


Figure 2.3: Overview of the RAG approach from [70]

In this context, the Retrieval Augmented Generation (RAG) approach [70], tries to formalize this inherent problem in LLM. As in Figure 2.3, the intrinsic nonparametric knowledge of LLMs is referred to as the knowledge acquired by the model during its training process and consists of the generation capabilities of the model from a pure linguistic perspective, and both on the semantic knowledge acquired during the training phase. However, the parametric knowledge is the external one, given as input to the model to enhance its generation capabilities.

More in detail, parametric knowledge can be implemented as a retriever from which relevant pieces of information, from the given input, are retrieved according to a similarity metric. Relevant information is then fed into the generative LLM along with the query, forcing the model as in an in-context learning approach with a context that dynamically adapts to the query [14, 70]. One of the most important

benefits of this method is the elimination of the costly fine-tuning process. Here, the parametric knowledge is the single module that can be customized to meet the target task, and the model, along its non-parametric knowledge, which can be only modified via fine-tuning, is left unchanged.

Here another significant paradigm shift can be found: the RQ is not towards the best fine-tuning strategy, but deeply relies on the suitable design of a retriever to determine the relevant context to inject customized knowledge into the model, and in the prompting strategy for the model itself. RAG seems to be an optimal trade-off for adapting large models, such as 70B and more, that cannot be fine-tuned both for computational constraint and for insufficient training data. Figure 2.4 shows a comprehensive and generic overview of a RAG architecture.

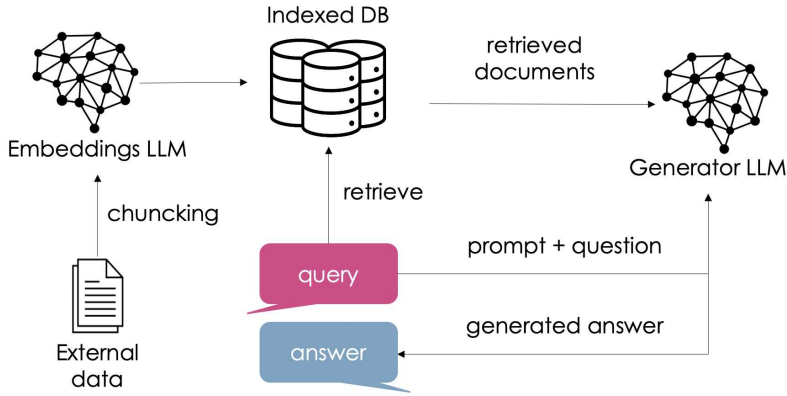


Figure 2.4: Generic RAG architecture

Starting from a documentary corpus $\mathcal{D} = \{d_1, d_2, \dots, d_n\}$ made up of n textual documents, each document is divided into small pieces, called chunks c , of a fixed token-length m , thus resulting $d_1 = \{c_1, c_2, \dots, c_l\}$, where l is the number of chunks in which d_1 is divided. Moreover, a small overlap of a variable amount of tokens for subsequent chunks is a common technique to provide a connecting context between chunks. Leveraging a suitable LLM trained for embedding generation (Embeddings LLM), textual chunks are converted into dense vectors $\mathbf{v}_i = \text{Embeddings_LLM}(c_i)$ that are stored in an indexed database. When a user query q is provided to the system, the embedded version of the query $\mathbf{v}_q = \text{Embeddings_LLM}(q)$ is calculated, so that the indexed Database (DB) is searched to retrieve the top k chunks. In particular, similarity measures such as cosine similarity or the Maximal Marginal Relevance [15] are calculated between the embeddings of the query and of the chunks in the indexed DB. The resulting scores are ordered and used as criteria to rank the elements in the database and the top k chunks that are the most similar to the given query are returned

as $top_k = \{c_{k_1}, c_{k_2}, \dots, c_{k_k}\}$. Finally, an appropriate prompt is designed with its instruction and filled with the textual form of the query and the retrieved documents,

$$prompt = instruction + query + top_k$$

the prompt thus obtained is then injected into a generative LLM (Generator LLM) that outputs an answer coherent to the given prompt:

$$answer = Generator_LLM(prompt)$$

The illustrated architecture (Figure 2.4) consists of various components, and each of them can be substituted with different implementation in different ways, e.g. vectorstore such as Facebook AI Similarity Search (FAISS) [29], Chroma [21] or Pinecone [103] can be used as indexed DB. Analogously, different options in the Embeddings or Generator LLM, both closed- and open-source models, along with the prompt design, can determine different behaviors and influence the performance of the overall system.

One of the main advantages of this approach is the in-context learning and instruction-following capabilities of the Generator LLM, which do not require a fine-tuning procedure. On this basis, the highly computational costs for both fine-tuning and building the appropriate large data set for fine-tuning itself are discarded, and the only required computational resources are the ones needed to infer the model.

Moreover, the adaptability of the RAG approach allows an elevated flexibility in knowledge management in the indexed DB, which can be easily modified with updated documents. In addition, the same architecture can be adapted to various domains, just with a prompt design and a suitable index DB.

In conclusion, the quality of the generated answers is high, since they are deeply influenced by the retrieved context that, if it is the correct one, forces in generating a correct answer with reference to the meaning. In addition, the explainability process over the generated answers is possible, since the availability of the context the answer is generated from.

RAG approach will be discussed, along with generative LLMs, in context of Unipa-GPT framework in the educational domain in Chapter 4, and in the DR-LLM framework as in the medical domain (Chapter 5).

Chapter 3

Encoder-only models were all we had

In this chapter are presented the works involving encoder-only LMs, used for traditional NLP tasks, namely Relation Extraction (RE) and semantic classification, which are evaluated with standard metrics such as the F1 score.

More in detail, the following RQ will be dealt in this Chapter:

RQ 1 *How to perform an English RE task and a semantic classification task in Italian in a resource-limited environment and utilizing encoder-only models?*

After a study of existing English data sets, and the related SOTA techniques for RE, SemEval 2010 Task 8 data set [45] was selected. It is a generalist data set and does not involve domain-specific knowledge or terminology, making it suitable for the experimentation phase. For this task, diverse configurations were tested that involve fine-tuning of different LMs and DNNs architectures were tested and two diverse encoding methodologies were proposed (Section 3.1).

In the context of EVALITA 2023 [66], two tasks in the Computational Ethics track were selected as case study for semantic classification task in Italian language.

EVALITA¹ is a periodic evaluation campaign of NLP and speech tools for the Italian language. Its objective is to promote the development of language and speech technologies for the Italian language by proposing different challenges through which different approaches can be evaluated. Two *minimalist* approaches, [127, 122] were proposed for Political Hate Speech Detection [65] in Hate Speech Detection (HaSpeeDe) (Section 3.2), and for the Homotransphobia detection task in Homotransphobia Detection in Italian (HODI) [95] (Section 3.3).

¹<https://www.evalita.it/>

Proposed pipelines leverages on Italian LMs, that are used as embedding generator model, while a DNN, such as Long Short Term Memory (LSTM), Bidirectional LSTM (BiLSTM), transformers and dense layers, were implemented for further processing of the extracted embeddings to obtain the final classification. Those developed approaches are referred to as *minimalist* since no further fine-tuning was performed over the selected LMs. In addition, a semantic analysis of the given data set provided useful insights in the context of the evaluation of the classification performances obtained.

3.1 Relation Extraction

Relation Extraction (RE) is a traditional NLP task, belonging to the main task of Information Retrieval (IR), in which, given a sentence s the objective is to extract the subject entity e_1 and the object entity e_2 , and determine if a semantic relation r exists among them, and classify it, thus obtaining the following triplet:

$$\langle e_1, r, e_2 \rangle$$

For example, given the following sentence

Irene was born in Italy

the triple expected to be extracted is

$$\langle Irene, birth_place, Italy \rangle$$

The subject entity e_1 can also be referred to the head h of the relation, while the object entity e_2 is the tail t .

Relations can be factual, as in the following sentence

Billy Mays, the bearded, boisterous pitchman who, as the undisputed
king of TV yell and sell, became an unlikely pop culture icon, died at
his home in Tampa, Fla, on Sunday.

the triplet $\langle Billy\ Mays \rangle, city_of_death, \langle Tampa \rangle$ connecting a person with a place, can be detected [154]. On other hand, more sophisticated, traversal and abstract relations can be considered such as the one connecting a component and its whole, as in

The ear of the African elephant is significantly larger—measuring 183
cm by 114 cm in the bush.

where the triplet $\langle ear \rangle$, *Component-Whole*, $\langle elephant \rangle$ can be identified [45].

In this work, a relation is considered to be a binary relation, i.e. a single head and the corresponding tail, despite RE tasks involving relations with n entities were also formulated [54]. In addition, RE task can be referred to sentence-level and document-level: in the latter, relations can occur both in the same sentence (as in sentence-level), both can be spread across different sentences that non necessarily are subsequent. In this work, binary sentence-level RE is investigated. Notwithstanding this broad definition, the RE task can be suitable to changes, depending on the specific data set: entities extraction, for example, can be limited to Named Entities or not, and the same entities can or cannot be annotated in the given text.

Even RE data sets are not unique as the process to create them, and in the resulting obtained relations. In this context, two principal strategies can be found: the former consists in the traditional human annotation, the latter is the so-called distant supervision [93]. High-quality data sets are undoubtedly obtained via human labeling, but this process is costly, and obtained data sets are quite small; on the other side, distant supervision allows an automatic labeling of large quantity of data, thus creating larger data sets that may contain a significant amount of noisy labels.

In the context of a preliminary study and in order to evaluate a baseline methodology and explore the usage of embeddings generated from traditional algorithms² and encoder-only LMs combined with DNN, the SemEval 2010 task 8 data set was selected [45].

3.1.1 SemEval 2010 Task 8

Semantic Evaluation ³ (SemEval) organizes yearly workshops in which English data sets are proposed for diverse challenges in NLP field. During the 5th International Workshop on Semantic Evaluation in 2010, the SemEval 2010 Task 8 was proposed [45], and it focuses on Multi-way classification of semantic relations between pairs of nominals.

It is a sentence-level RE data set consisting of 10717 sentences, each manually annotated according to 9 different predefined relations, plus the *Other* relation. Selected relations are mutually exclusive, thus an entities pair only maps to a single relation, and were designed to not overlapping each other and being general and not refer to any specific domain. Statistics of the data set are reported in Table 3.1.

²Traditional here refers to techniques that not involve transformers

³<https://semEval.github.io/>

Table 3.1: Statistics of SemEval 2010 Task 8 data set

Relations	Training	Validation	Test	Total
Cause-Effect	847	156	328	1331
Instrument-Agency	419	85	156	660
Product-Producer	603	114	231	948
Content-Container	483	57	192	732
Entity-Origin	614	102	258	974
Entity-Destination	832	13	292	1137
Component-Whole	870	71	312	1253
Member-Collection	682	8	233	923
Message-Topic	629	5	261	895
Other	1130	280	454	1864
Total	7109	891	2717	10717

The proposed task consists of correct classifying the existent relation among the two annotated entities, also taking into account the direction of the extracted relation. In the data set, in fact, the first entity in the sentence is marked as $e1$ and the second as $e2$.

The objective of the challenge is not only to determine that *relation3* occurs between the two entities, but also to determine whether the associated triple is $\langle e1, relation_3, e2 \rangle$ rather than $\langle e2, relation_3, e1 \rangle$. In this context, the entire task may be seen as a classification task of the sentences and pairs of entities provided in 19 different classes: the original 9 relations are duplicated as *relation(e1,e2)* and *relation(e2,e1)* thus highlighting the direction of the relation as a different relation, plus the *Other* relation.

Given the following pairs of *Entity-Destination* relation,

$\langle e1 \rangle$ People $\langle /e1 \rangle$ have been moving back into $\langle e2 \rangle$ downtown $\langle /e2 \rangle$.
Entity-Destination(e1,e2)

The stainless-steel $\langle e1 \rangle$ vessel $\langle /e1 \rangle$ was filled with $\langle e2 \rangle$ nitrogen $\langle /e2 \rangle$,
and the pressure in the outer volume of the chamber was kept steady.
Entity-Destination(e2,e1)

it is possible to notice how in the first sample, the first annotated entity coincides with the head of the relation, while in the second one, the first one is the tail. Correct classification, then, relies on both relation and direction identification.

As for the *Other* relation, is used to evidence the presence of a semantic relation between the target entities in the given sentence, but its typology is not further specified. In the following sample

He produces drawings and videos that have been shown in museums
in Europe and America

a relation can be intuitively found between *drawings* and *museums* but nor its type and direction cannot be further determined [45].

3.1.2 Related works

An analysis in the SOTA methods in RE was conducted, with focus on the approaches that were tested in the target data set SemEval 2010 Task 8. A summarization work on the methods analyzed, shown as a common pipeline, reported in Figure 3.1 in which three main modules can be identified, namely a feature extractor, a feature assembler and a classifier. Although the last module remained unchanged throughout the years, the feature extractor and feature assembler changed greatly with increasing progress in DL research, thus influencing the various proposed methodologies.

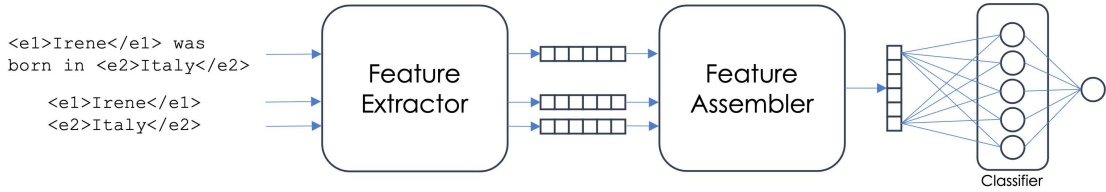


Figure 3.1: Overview of a generic Relation Extraction pipeline

Dealing with textual data, a primary necessary step in pre-processing is required in terms of embedding representation, that is, a dense numeric representation of a given word. A step further was made for a one-hot or frequency-based representation thanks to non-contextual embeddings, dense vectors that condensed the meaning of a word into a vector of fixed length, as for GloVe [102], Word2Vec [90] or fastText [40]. This vector representation is usually combined with other features involving the location of a word in a phrase (positional embeddings), or its POS tag. Following a chronological order, the embeddings and associated features, where combined via a feature assembler such as a Convolutional Neural Network (CNN) [68, 149].

In addition to the inherent linear form of textual content, sentences can also be represented as a tree structure, called dependency tree, in which grammar dependencies among words are encoded as directed arcs labeled from head words

to dependent ones [58]⁴. In this context, native neural networks to the graph structure are preferred, such as Graph Neural Network (GNN) [116] and its convolutional variant Graph Convolutional Network (GCN) [60]. Both architectures were involved for RE task with GloVe embeddings [159, 41] and in conjunction with BiLSTM [153, 128].

The paradigm change in NLP with the attention mechanism [137], allowed an important shift in embeddings representation, moving from non-contextual embeddings towards a contextual representation (Section 2.1). This new embedding representation was used both in the neural classifier in addition to linguistic features [6, 140, 131, 71] but also in combination with GNN [155], GCN [156] and GCN combined with BiLSTM [86].

In conclusion, regardless of the chosen sentence representation, the extracted features and the DL method to process the extracted feature, RE task is essentially a classification task, where a suitable classifier, made up of a dense layer and an appropriate activation function, ends the architectures.

3.1.3 Methods

In this Section, the proposed encoding and network architecture for the RE task with SemEval 2010 task 8, are reported.

3.1.3.1 Encoding

In this context, some attempts were conducted in terms of RE pipeline in which two different encodings of the given sentence were tested. From a sample of the data set, (Figure 3.2.a), for Textual Encoding (TexEnc), sentence and entities are isolated and reported separately, thus obtaining their three textual form (Figure 3.2.b), while in the Concatenated Encoding (ConEnc), a unique textual sample is built via a concatenation of both sentences and entities separated by a separation token [SEP] (Figure 3.2.c).

From the proposed encoding, different PTLMs were considered as encoders for embeddings calculation, namely BERT [27] in the base cased version (BERT-c) and uncased version (BERT-uc), base cased SpanBERT [56], RoBERTa base [81], XLM-RoBERTa base [23]. Note that both RoBERTa and XLM-RoBERTa, which are used in their cased version, natively uses `</s>` as separation token, then this token is inserted for those models instead of [SEP], which is supported as separation token for the remaining models.

⁴p. 414

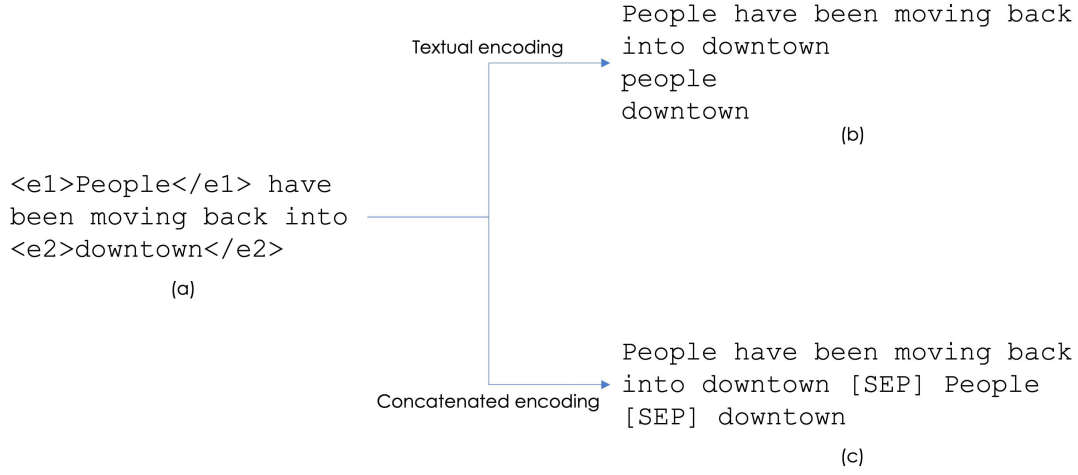


Figure 3.2: Example of encodings from a sample of SemEval 2010 Task 8 data set (a), namely textual (b) and concatenated (c) encoding

3.1.3.2 Network architecture

Given the two encodings, two different types of architecture were developed to better fit the cured sample. As for TexEnc, the three extracted elements were injected separately into a LM to extract the three corresponding [CLS] embeddings. While in the ConEnc, the entire sequence is fed in a LM to extract the embeddings of [CLS]. The former representation will be three times the lengths of the latter representation, which is equal to the length of the embeddings produced by the selected LM. As for BERT, [CLS] embeddings have a dimensionality equal to 768, this will be the length of the embeddings generated after ConEnc, while embeddings obtained by TexEnc will be 2304 long.

Both representations provide the same information to the model for the subsequent relation classification in a different manner. In TexEnc, in particular, a simulated non-contextual embedding is extracted from a contextual models, via directly injecting single entities, without any context, in the LM. On the contrary, in ConEnc, a single embedding containing both the target sentence and the target entities is extracted using the contextual aspects of the embedding generation process in encoder-only architectures.

Obtained embeddings are then elaborated via a custom neural layer called *DNDR*, which combines a Dense linear layer, followed by a batch Normalization, Dropout, and a ReLU activation function. While for embeddings generated via ConEnc, the hidden dimension is kept in DNDR elaboration, in TexEnc, embeddings dimensions are reduced by one third and thus are equal to ConEnc ones. In conclusion, a 19-classifier ends the architecture. The proposed architectures are

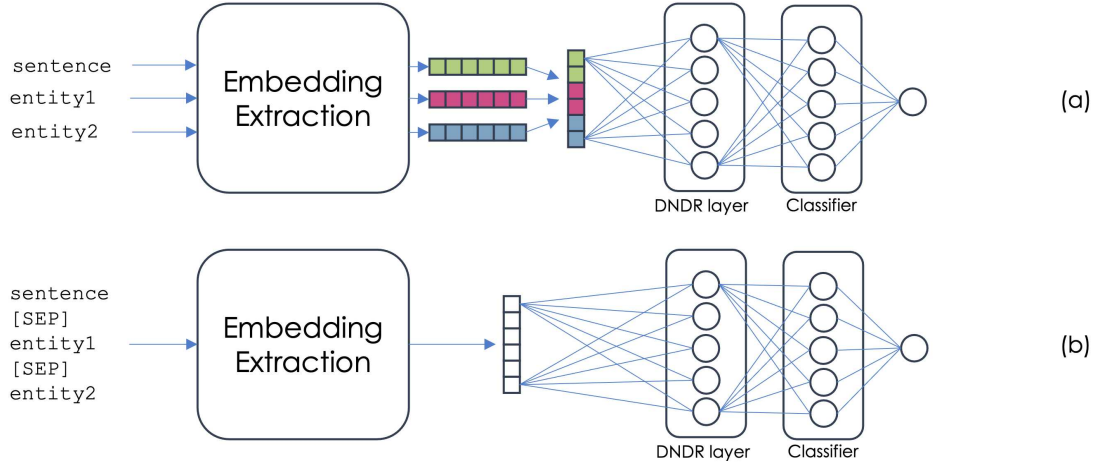


Figure 3.3: Overview of the proposed architecture for SemEval 2010 Task 8 for textual encoding (a) and concatenated encoding (ConEnc) (b)

illustrated in Figure 3.3.

Regarding the experimental phase, both architectures were trained with two different strategies: one is the end-to-end fine-tuning of the entire pipeline, including the selected LM, while the other is only directed toward the DNR layer and the classifier, using the original training of the selected PTLN. The former training strategy will be named model-FT-DNR, while the latter is named model-DNR, in both cases, the model is a placeholder for the used PTLN. The training process was performed on a machine equipped with two Intel Xeon E5 CPUs 96GB RAM and a NVIDIA TITAN Xp GPU 12GB RAM. Diverse hyperparameters were considered for the learning rate and dropout values, resulting in optimal results with a dropout value of 0.2, Adam optimizer [59] with a learning rate of 0.0001, batch size of 32 and Binary Cross Entropy (BCE) loss for 300 epochs, with a patience value calculated over validation loss equal to 25.

3.1.4 Results

The experimental results on the test set, calculated through the official scorer on the test set, which is a macro-averaged F1 score for the (9+1) way classification, taking into account directionality[45], are reported in Table 3.2.

Best performances were obtained via ConEnc combined with a LM fine-tuning and DNR layer along with a classifier, in particular RoBERTa reaches 85.45, the highest F1 score. Other models with the same architecture are placed a little bit below with values raging 83 and 84. Overall ConEnc places above TexEnc and

Table 3.2: Macro-averaged F1-Score for (9+1)-way classification over the test set of SemEval 2010 Task 8 with the proposed architectures.

	model-FT-DNDR		model-DNDR	
	<i>TexEnc</i>	<i>ConEnc</i>	<i>TexEnc</i>	<i>ConEnc</i>
BERT-c	79.48	84.57	<i>70.92</i>	58.75
BERT-uc	80.97	84.27	69.28	<i>60.65</i>
SpanBERT	<i>82.71</i>	84.24	67.81	59.95
RoBERTa	77.41	85.45	66.02	57.63
XLM-RoBERTa	78.51	83.84	64.53	56.61

models highly benefit of LM fine-tuning. No predominant LM is found to be much more performative than the others, on the contrary, XLM-RoBERTa reaches the lowest performances despite different architectures and encodings. As for ConEnc, the presence of the separation token allows different LMs to better process the tokens provided, resulting in much more effective concatenation in contrast to concatenation as in TexEnc. In addition, despite the specific LM, all the architectures tend to cluster in separate ranges, as for results and similar behaviors can be found in their training process. In fact, all models tend to learn in the first few epochs, after which training loss decreases approaching overfitting and validation loss starts increasing, showing high generalization errors in the model. The main reason behind this behavior lies in the choice of the LMs themselves: as they are PTLMs, an significant pre-training was performed over a huge amount of data and the exiguous cardinality of the considered training set is not enough to move weights in the model with and without fine-tuning the LMs.

To further stress the developed architectures, some variants were tested, in particular with the removal of the DNDR layer. As for the models with LM fine-tuning, removal of the DNDR layer did not have a significant impact: the performances are slightly lower but still remain in the same range. The same happens as for the classifier that uses frozen LMs for embedding extraction: in this case the main difference is an increasing number of epochs needed to activate the early stopping based on the validation loss, but the overall obtained results still place over to the equivalent model with the DNDR layer. Generally speaking, the DNDR is a layer that helps in lowering the training time of the models and increasing their classification performances.

Table 3.3: Macro-averaged F1-Score for (9+1)-way classification over the test set of SemEval 2010 Task 8 with the SOTA models and the best proposed model.

SOTA model	SemEval 2010 Task 8
RIFRE [155]	91.30
REDN [71]	91.00
Skeleton-Aware BERT [131]	90.36
EPGNN [156]	90.20
BERTEM+MTB [6]	89.50
Entity-Aware BERT [140]	89.00
C-GCN-MG [86]	85.90
Proposed model	85.45

With respect to the performances of SOTA models on this task⁵ in Table 3.3, the proposed strategy did not represent a significant improvement in literature, in which the F1 score ranges from 86 to 92, as for recent developed models.

⁵<https://paperswithcode.com/sota/relation-extraction-on-semeval-2010-task-8>

3.2 HaSpeeDe3

The continuous spread and usage of social networks has become a problem when dealing with hate online. All social platforms use artificial intelligence techniques to detect and report or remove some dangerous content in terms of hate or violence. The interest in this respect is also high in the scientific community; in fact, different international campaigns for detecting hateful speeches have been proposed in recent years: OffensEval [147, 148], HatEval [9], HaHackathon [87].

Detection of hateful content in Italian has been addressed by the HaSpeeDe evaluation competitions, which in 2023 reached its third edition [65]. Unlike the two previous editions [13, 114], it explores hate speech in strong polarized debates, concerning politics and religion. The objective of Task A, “Political Hate Speech Detection Textual”, is to predict hate speech in-domain settings, using only the provided textual content of the tweet, will be the focus of this Section.

To address this challenge, the proposed approach relies on encoding the text into suitable word embeddings that are processed via neural architectures like LSTM, BiLSTM or transformers. The generated output is further processed by a classifier to detect the presence of hateful content. The proposed pipelines will be referred to as *minimalist* architecture. Moreover, on purpose, it was the choice not to fine-tune the selected PTLMs, to stress the use of light networks to be trained with low computing resources.

In this Section a description of the systems along with data pre-processing is reported (Section 3.2.1) and the results are reported and discussed in Section 3.2.2.

3.2.1 Methods

The focus of HaSpeeDe3 is on political and religious hate, where strong polarized opinions can be found. The data set used in this edition for task A is the PolicyCorpus XL [16] that contains 7000 manually annotated tweets, of which more than 40% are labeled hateful (H). The training data set was released for the campaign with a total of 5600 samples: for developing purposes, the given data set was randomly split into a training and validation set, using a 80-20 ratio, resulting in 4480 and 1120 samples respectively.

3.2.1.1 Pre-processing

The [URL] tag, reference to mentions and retweet notes were removed since they were not considered meaningful: in particular, mentions are referred to anonymized accounts and thus do not add special information. This was done after an analysis

of the most cited words and hashtags, in which all words were reported in their lower case form. As reported in Table 3.4, the [URL] tag is the most frequent between classes and does not add information just like anonymized mentions in the form @unknown. In general, no other relevant words appeared that suggest a strong separation between classes. The same considerations can be made for the hashtags as reported in Table 3.5.

Table 3.4: Word distribution statistics over HaSpeeDe3 data set divided per label hateful (H) and not-hateful (NH).

All tweets	freq	NH tweets	freq	H tweets	freq
url	1585	url	1007	url	578
governo	455	unknown	297	governo	160
solo	376	governo	295	solo	133
unknown	364	solo	243	797998657209770	132
fare	276	fare	178	salvini	116
fatto	264	fatto	173	fa	112
fa	256	oggi	165	sempre	104
cosa	254	essere	159	cosa	99
salvini	247	cosa	155	fare	98
essere	247	italia	152	poi	96

Although there are some hashtags that are hateful (such as *salvinipagliaccio*, *speranzadimettiti* and *governodeipeggiori*), the most frequent ones are just either politicians’ or parties’ names, and politics related words, that do not express any polarized content. Moreover, since a strong and significant distinction between hateful and non-hateful hashtags can be made, their information has been used as a word inside the tweet, thus keeping the crucial information, while the hashtag symbol was removed.

Similar considerations were made for emojis: also in this case, a strong polarization in the use of emojis did not arise, particularly for those that are more associated with disgust and hate (Table 3.6). Since emojis are deeply used in social media communication, they were kept. No further elaboration was made on the tweets: words were not reported in their lower case form, allowing a more accurate extraction of embeddings for the case-sensitive PTLMs. Regarding emojis, uppercase texts have a specific meaning in social media communication in terms of prosodic and emotional interpretation [44, 17].

Table 3.5: Hashtags distribution statistics over HaSpeeDe3 data set divided per label hateful (H) and not-hateful (NH).

All tweets	freq	NH tweets	freq	H tweets	freq
salvini	886	salvini	557	salvini	329
m5s	630	m5s	495	salvinisciacallo	251
conte	419	conte	316	salvinipagliaccio	235
draghi	368	legge	284	governodeipeggiori	223
lega	341	governo	255	speranzadimettiti	179
governo	320	draghi	252	speranzavattene	175
legge	309	lega	218	m5s	135
renzi	306	renzi	212	salviniportasfiga	131
salvinisciacallo	255	pd	181	lega	123
pd	251	politica	170	draghi	116

Table 3.6: Emoji distribution statistics over HaSpeeDe3 data set divided per label hateful (H) and not-hateful (NH).

All	freq	NH	freq	H	freq
	38		21		32
	32		15		27
	32		14		25
	26		13		18
	26		11		15
	26		10		11
	22		8		10
	15		8		10
	15		8		8
	13		7		7

3.2.1.2 Network architecture

Different models were developed that share the same macrostructure shown in figure 3.4. The key idea was to stress, as much as possible, existent neural architectures for sequence processing, that are LSTM [46], BiLSTM and transformers [137]. Those architectures serve as a Neural Models that are used to further

process the extracted embeddings.

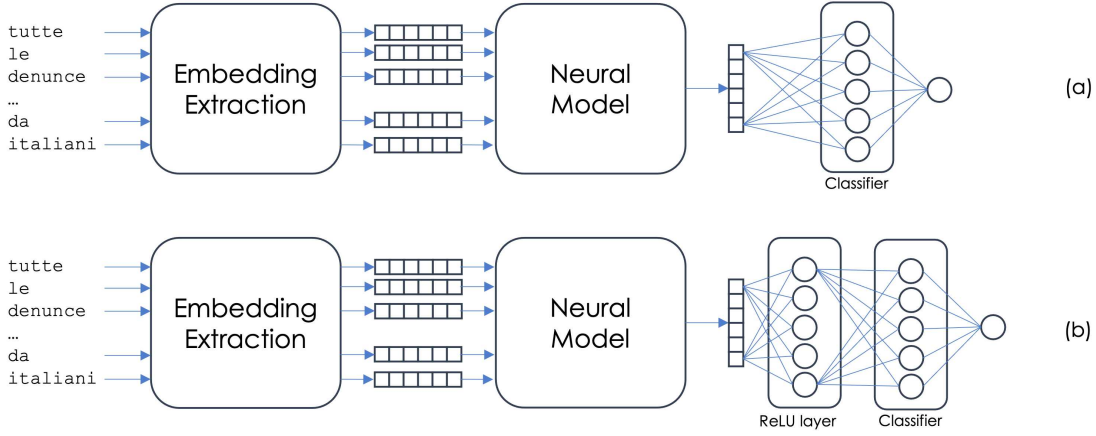


Figure 3.4: Overview of the proposed architectures for HaSpeeDe3, (a) is made up of a module for embeddings extraction, a neural module and a classifier, while (b) adds an additional ReLU dense layer.

After pre-processing, the input sentences were padded to $maxLength+2$ tokens where $maxLength$ is the size of the longest sentence, and the remaining two tokens are respectively the [CLS] and the [SEP] one. Either a PTLM or a static context-free embedding model were used for embedding generation. As for Encoder-based LMs, embeddings with a dense dimension of 768 were extracted from the last layer of the following PTLMs: BERT base multilingual cased (mBERT) [27], BERT base italian uncased (BERT-it) [118], XLM-RoBERTa [23] and ALBERTo [104]. Those models were used in their base form provided by the HuggingFace transformer library⁶. Either fastText [40], a context-free embedding model, was considered in this analysis, to extract 300-long embeddings. Both PTLMs and fastText were used in their base form and no fine-tuning was performed for embedding extraction. In the context of building a *minimalist* model, fine-tuning the selected PTLMs is an option that is not taken into account, since the main idea is to stress the use of light networks to be trained with low computing resources. Moreover, only XLM-RoBERTa, fastText, and mBERT generate case-sensitive embeddings.

The generated embedding is fed into a Neural Module for feature extraction that consists of a LSTM or a BiLSTM or a transformer: henceforth, the developed architectures are named according to the specific neural module. The output feature vector has the same size as the word embedding with the exception of the BiLSTM that generates a double-length output. Finally, the feature vector is passed to a classifier made by either 300 or 768 linear units, depending on the

⁶<https://huggingface.co/docs/transformers/index>

length of the embedding, and a sigmoidal output to achieve binary classification (Figure 3.4.a). Some experiments were run by inserting a ReLU dense layer before the aforementioned one, made up of a dense linear layer and a ReLU. Those architectures are referred as LSTM-Deep, BiLSTM-Deep and Trasformer-Deep (Figure 3.4.b).

The illustrated architectures were trained only on the given data set using a machine equipped with two Intel Xeon E5 CPUs 96GB RAM and an NVIDIA TITAN Xp GPU 12GB RAM. Hyperparameters were selected as follows: dropout values in $\{0.1, 0.2\}$, batch size 32, Adam optimizer [59] with learning rate 0.01, and a BCE loss. Selected models were trained for a maximum of 1000 epochs with a patience value of 50.

Different feature extractors were implemented using 1, 2 or 3 LSTM, BiLSTM or transformer layers, but the best results were obtained from the single-layer feature extraction modules. In addition, the developed models are relatively small, where the trainable parameters range from 1M to 10M.

3.2.2 Results

Table 3.7: Overview of all the relevant results across the developed architectures on the golden labels from the HaSpeeDe3 test set. Starred results are the ones submitted to the competition, while the bold one is the highest score. Underlined results were obtained by removing stopwords from the data.

	LSTM	LSTM-Deep	BiLSTM	Transformer-Deep
XLM-RoBERTa	<u>0.84</u>	<u>0.82</u>	0.88	0.84
fastText	<u>0.79</u>	<u>0.73</u>	<u>0.85*</u>	<u>0.86</u>
AlBERTo	0.86	0.83	0.83*	<u>0.85</u>
mBERT	0.86	<u>0.83</u>	0.86	0.83
BERT-it	0.85	<u>0.82</u>	0.87	<u>0.80</u>

The best F1-macro performances obtained on the test set from the developed models are reported in Table 3.7. The submitted modes were the best runs with respect to the validation set, namely AlBERTo w/ BiLSTM (run 1) and fastText w/ BiLSTM (run 2). After the release of the golden labels, it was possible to measure the actual performance of all developed systems, and this shows that the XLM-RoBERTa w/ BiLSTM architecture gives the best results, ranking 7th on the leaderboard, while submitted runs are at the last place, as shown in Table 3.8. All other participants to the challenge proposed models that involved fine-tuning

of the used LM, which resulted in a most performative strategy. In contrast, the task organizers proposed a simple baseline based on a Support Vector Classifier trained with a unigram representation of the textual content of the tweet, which is lower than the average of the proposed submitted models [65].

Table 3.8: Macro F1 results over HaSpeede3 test set of the submitted models and the actual best one (starred). Both baseline result, ranking and expected ranking position are reported.

Run name	Macro F1	Rank
XLM-RoBERTa w/ BiLSTM*	0.88*	7*
fastText w/ BiLSTM	0.85	8
<i>Baseline</i>	0.85	9
ALBERTo w/ BiLSTM	0.83	10

Best results are obtained either when using a PTLM coupled with a LSTM and a BiLSTM as feature extractor and a single dense layer⁷, while the transformer-based networks exploit better a context-free embedding by using a two layer classifier.

Models were tested both with and without stopwords in the inference data set (underlined values in Table 3.7), but only the models that use fastText benefit from it, while the PTLMs perform almost equally over LSTM and BiLSTM configurations as expected. In the training phase, ALBERTo outperformed the other PTLMs because it uses a more accurate tokenization compared to the others and takes advantage of its inner knowledge: ALBERTo was trained on a corpus of Italian tweets that share the same linguistic macrostructure as PolicyCorpus. Both PolicyCorpus and ALBERTo training sets are derived from social media content, which results similar within each other for their linguistic style, usage of emoji and social slang. On the other hand, the best model is the one based on XLM-RoBERTa: this can be caused by its tokenizer, which owns an inner representation for emojis, and consider them as unique tokens and not as [UKN].

3.2.2.1 Error analysis

In addition to the aforementioned differences between the PTLMs used, another analysis was made on the misclassified tweets by comparing the results of the best architectures (ALBERTo w/ LSTM, ALBERTo w/ LSTM-Deep, XLM-RoBERTa

⁷In Table 3.7 some experiments and configurations are not reported, like the BiLSTM-Deep one, because they ran badly with respect to the submitted architectures.

and fastText w/Transformer-Deep) and the submitted models. The models agree to misclassify 32 tweets, and 25 of them are labeled hateful.

None of these misclassified tweets contain emoji, that is, their presence or absence is not source of bias in those models. Moreover, the majority of these tweets contain hashtags or expressions referring to politicians and topics of interest in the political debate, that *per se* are not hateful. On the contrary, tweets containing the hashtag *speranzadimettiti*, considered hateful as in Table 3.5, can be found in nonhateful tweets, as in the one below:

#Drittoerovescio Italia tra #coprifuoco e #lockdown ... #Speranzadimettiti [URL]

In this and similar tweets, the author disapproves the governmental behavior of a minister: in this case, it cannot be considered as hateful since it expresses a negative opinion without insulting.

On the other side, hateful tweets usually contain profanities and vulgar expressions: hateful tweets that are not correctly classified by the developed models, lack of those expressions or put them in an unconventional way (self-obfuscation or embedded in other words) and this lead to their mis-classification.

3.3 HODI

The increasing interest for gender-inclusive and nondiscriminatory language passes through its counterpart in hate speech that it is largely spreading on social networks, particularly against the LGBTQIA+ community. NLP community is involved in developing systems for the detection of hate speech as in MAMI (Multi-media Automatic Misogyny Identification) [32] and EDOS (Explainable Detection of Online Sexism) [61] where the focus was on the detection of misogyny and sexism, but these data sets are not focused on Italian nor on the detection of hate speech against people from the LGBTQIA+ community.

HODI [95] is the first shared task for the automatic detection of homotransphobia in Italian, and subtask A, detection of homotransphobia, focuses on the binary textual classification of homotransphobic tweets. More in detail, homotransphobic content is content that expresses hatred, disapproval, and fear toward people based solely on their gender identity and/or sexual orientation.

To address this challenge, the proposed approach relies on encoding the text into suitable word embeddings that are processed via neural architectures like LSTM, BiLSTM or transformers. The generated output is further processed by a classifier to detect the presence of homotransphobic content. The proposed pipeline will be referred to as *minimalist* architecture. Moreover, on purpose, it was the choice not to fine-tune the selected PTLMs, to stress the use of light networks to be trained with low computing resources.

In this Section a description of the systems along with data pre-processing is reported (Section 3.3.1) and the results are reported and discussed in Section 3.3.2. Note that this Section contains examples of potentially offensive content, whose profanities have been obfuscated with PrOf ⁸ [96].

3.3.1 Methods

The data set by the HODI organizers contains 6000 Italian tweets annotated accordingly to the presence of homotransphobic content. Since the training set released for the competition was made up of 5000 samples, this was randomly split into a training and validation set, using a 80-20 ratio, resulting in 4000 and 1000 samples, respectively.

Table 3.9: Word distribution statistics over the HODI data set divided per label.

All tweets	freq	NH tweets	freq	H tweets	freq
culo	1283	culo	937	r*cchione	596
url	995	rotto	657	ch*cca	488
r*cchione	971	url	637	url	358
rotto	912	gay	544	culo	346
gay	700	c*zzo	398	rotto	255
ch*cca	688	r*cchione	375	gay	156
c*zzo	529	fare	328	isterica	151
fare	466	caghino	287	fare	138
solo	348	solo	231	c*zzo	131
me	310	me	211	quel	131

3.3.1.1 Pre-processing

The [URL] tag, reference to mentions and retweet notes were removed since they were not considered meaningful: in particular, mentions are referred to anonymized accounts and no special information is added. This was done after an analysis of the most cited words and hashtags, in which all words were reported in their lower case form. As reported in Table 3.9, the [URL] tag is the most frequent between classes and does not add information just like anonymized mentions, which are not reported. During this analysis, it was interesting to note that the most cited words are slurs directed at LGBTQIA + members. Although a first idea for approaching the task was to look for slurs, direct inspection of the data set clearly shows that slurs are not a good indicator of homotransphobic content. Slurs, in fact, are widely used by LGBTQIA + people as a self-definition method suggesting a (re-)appropriation of the term itself [94], and obviously tweets of this kind cannot be considered homotransphobic, so the slur word loses its negative connotation, as in the tweet:

firmato una fr*cia in sessione:(⁹

Here, the term *fr*cia* does not have any negative connotation. Therefore,, any word-dependent consideration about the polarization of homotransphobic speeches cannot be made, as the presence of slur words does not convey negative content, i.e. slurs cannot be regarded as representative elements for separating classes.

⁸<https://github.com/dnozza/profanity-obfuscation>

⁹signed by a queer during the exam session:(

The same considerations hold for the hashtags as reported in Table 3.10 where the most frequent are neutral words. Hence, the hashtag symbol was removed and the subsequent word was kept along with its meaning inside the tweet.

Table 3.10: Hashtags distribution statistics over HODI data set divided per label.

All tweets	freq	NH tweets	freq	H tweets	freq
gay	15	pride	14	gioelemagaldi	9
pride	14	prelemi	13	conte	6
prelemi	14	eurovision	11	casalino	5
eurovision	13	gay	10	gay	5
tellonym	9	tellonym	8	tortura	5
gioelemagaldi	9	pridemonth	8	attacchi	4
draghi	8	dazn	7	draghi	4
pridemonth	8	meloni	6	biohacking	3
dazn	7	omofobia	5	fronte	3
conte	7	escita	5	intelligence	3

Table 3.11: Emoji distribution statistics over HODI data set divided per label.

All	freq	NH	freq	H	freq
😂	95	😂	45	😂	50
🤔	83	🤔	44	🤔	39
😭	23	😭	19	😭	13
❤️	19	❤️	15	😂	9
👍	19	👍	13	👍	8
😂	17	👍	11	🌈	8
🤢	16	😭	9	!	8
🤔	15	🤢	9	😂	7
😂	14	😭	8	🤢	7
😭	13	👍	8	🤔	6

Similar considerations were made for emojis: also in this case a strong polarization is not found in the use of emojis, in particular in those that are more associated with disgust and hate (Table 3.11). Since emojis are deeply used in social media communication, they were kept. Based on the statistics reported

in Table 3.11, the emoticons contained in the data set were manually substituted with the corresponding most frequent emoji. As an example, the “:(” emoticon was translated in “😞” even if this is not the exact correspondence. This approach does not inject bias in the data set as the different emoticons were very few, while their rough meaning is preserved, thus avoiding to consider them as mere sequences of punctuation marks. No further elaboration was made on the tweets: words were not reported in their lower case form, allowing a more accurate extraction of embeddings for the case-sensitive PTLMs. Regarding emojis, uppercase texts have a specific meaning in social media communication in terms of prosodic and emotional interpretation [44, 17].

3.3.1.2 Network architectures

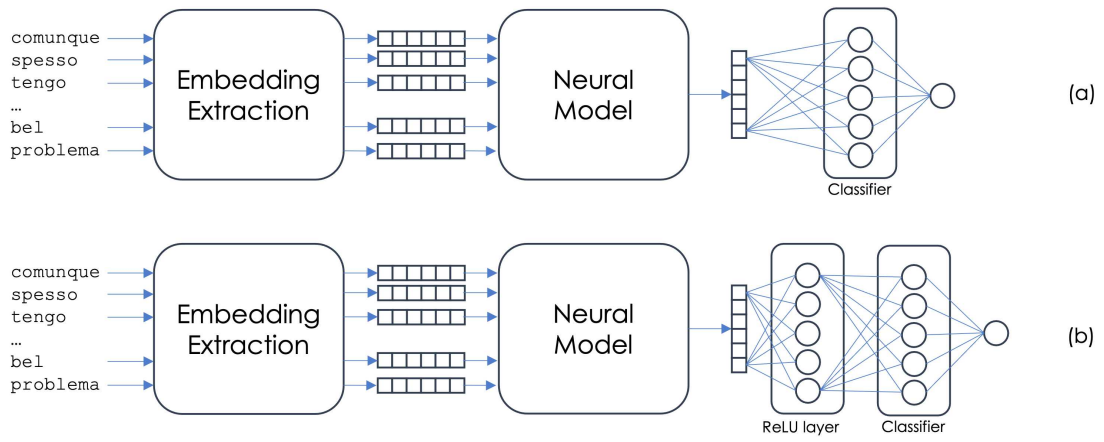


Figure 3.5: Overview of the proposed architecture for HODI (a) has a module for embeddings extraction, a neural module and a classifier while (b) adds an additional ReLU dense layer.

Different models were developed that share the same macrostructure shown in Figure 3.5. The key idea was to stress, as much as possible, existent neural architectures for sequence processing, that are LSTM [46], BiLSTM and transformers [137]. Those architectures are used to further process the extracted embeddings.

After pre-processing, the input sentences were padded to $maxLength+2$ tokens where $maxLength$ is the size of the longest sentence, and the remaining two tokens are respectively the [CLS] and the [SEP] one. Either a PTLM or a static context-free embedding model were used for embedding extraction. As for Encoder-based LMs, embeddings with a dense dimension of 768 were extracted from the last layer of multilingual cased BERT base (mBERT) [27], Italian uncased BERT (BERT-it) [118], XLM-RoBERTa [23] and ALBERTo [104]. Either fastText [40], a context-

free embedding model, was considered in this analysis, for extracting 300-long embeddings. Both PTLMs and fastText were used in their base form and no fine-tuning was performed for embedding extraction. In this context, fine-tuning is an option that is not taken into account, since the main idea is to stress the use of light networks to train with low computing resources. Moreover, only XLM-RoBERTa, fastText, and mBERT generate case-sensitive embeddings.

The extracted sequence of embeddings is fed into a neural module that consists of a LSTM or a BiLSTM or a transformer: henceforth, the developed architectures are named according to the specific neural module. The output feature vector has the same size as the word embedding with the exception of the BiLSTM that generates a double-length output. Finally, the feature vector is passed to a classifier made by either 300 or 768 linear units, depending on the length of the embedding, and a sigmoidal output to achieve binary classification (Figure 3.5.a). Some experimental configurations add an extra ReLU dense layer before the aforementioned classifier, made up of a dense linear layer and a ReLU. Those architectures are referred as LSTM-Deep, BiLSTM-Deep and Trasformer-Deep (Figure 3.5.b).

The illustrated architectures were trained only on the given data set using a machine equipped with two Intel Xeon E5 CPUs 96GB RAM and an NVIDIA TITAN Xp GPU 12GB RAM. Hyperparameters were selected as follows: dropout values in $\{0.1, 0.2\}$, batch size 32, Adam optimizer [59] with learning rate 0.01, and a BCE loss. Models were trained for a maximum of 1000 epochs with a patience value of 50.

Different feature extractors were implemented using 1, 2 or 3 LSTM, BiLSTM or transformer layers, but the best results were obtained from the single-layer feature extraction modules. In addition, the developed models are relatively small, where the trainable parameters range from 1M to 10M.

3.3.2 Results

The best models during the evaluation window, according to their performance in the validation set, were BERT-it w/ Transformer (run 1), AlBERTo w/ TransformerDeep (run 2) and AlBERTo w/ LSTM (run 3) and placed at the bottom of the rank and below the baseline [95]. Due to an internal error in the code of the training procedure, the submitted results are intrinsically wrong, and for this reason, all the experiments were repeated using the correct architecture, after the release of the golden labels. An overview of all the models developed is reported in Table 3.12, while Table 3.13 shows the runs submitted, their fixed counterpart, and the baseline value. In both tables the results refer to the F1-macro score over the test set and, although all possible configurations were run, in Table 3.12 the significant architectures are reported, i.e. the configurations that were placed above the baseline, which involves the usage of a Logistic Regression classifier based on

Table 3.12: Overview of the best relevant results across the developed architectures on the golden labels from HODI test set. Italic values place below the baseline, while the bold one is the highest score. Underlined results were obtained by removing stopwords from the data.

	LSTM-Deep	BiLSTM	Transformer	Transformer-Deep
XLM-RoBERTa	<i>0.57</i>	0.68	<i>0.65</i>	0.68
fastText	<u>0.55</u>	<u>0.70</u>	0.68	0.68
AlBERTo	0.72	0.75	<u>0.71</u>	0.75
mBERT	<i>0.33</i>	<i>0.63</i>	<i>0.34</i>	0.73
BERT-it	<u>0.64</u>	0.70	0.73	<u>0.69</u>

TF-ID using only unigrams and bigrams [95].

The results show that the AlBERTo w/ Transformer architecture with a two dense layers classifier (Transformer-Deep) has the best performance, and it is expected to rank at the 10th place on the leaderboard. Other task participants proposed different approaches including the usage of LLMs and its fine-tuning [95].

Table 3.13: Macro F1 results over HODI test set of the submitted models and their fixed versions (starred). Both baseline result, ranking and expected ranking position are reported.

Run name	Macro F1	Rank
AlBERTo w/ TransformerDeep*	0.75	10*
AlBERTo w/ BiLSTM*	0.74	11*
BERT-it w/ Transformer*	0.74	13*
<i>Baseline</i>	0.67	13
AlBERTo w/ BiLSTM	0.55	17
BERT-it w/ Transformer	0.52	18
AlBERTo w/ TransformerDeep	0.52	19

Moreover, LSTM-Deep and BiLSTM models exhibit comparable performance: bi-directional sequence processing compensates for the reduced classifier’s capacity. In general, the Transformer-Deep architectures performed better than the Transformer ones.

As expected, only fastText-based models benefit from removing stop words. The models using ALBERTo and BERT-it achieved almost the best results in both the training phase and the evaluation, because the network can take advantage of PTLMs that are specifically fine-tuned in the target languages. In particular, ALBERTo was trained on a corpus of Italian tweets that share the same linguistic macro-structure of the data set proposed for HODI competition. Both HODI and the ALBERTo training set are derived from social media content, which results similar to each other for their linguistic style, use of emoji and social slang.

3.3.2.1 Error analysis

An error analysis was performed particularly on tweets that were mis-classified by the models reported in Table 3.12 that perform better with reference to the baseline. All classifiers incorrectly agreed on 40 tweets: 80% of them were homotransphobic. Due to a direct analysis of their content, the following consideration can be made.

The first consideration is that most misclassified tweets contain slurs. As shown in Section 3.3.1.1, slur words are widely used by LGBTQIA + people as self-reference without any discriminatory intent, so an automatic classifier may not recognize these shades of meaning as in:

Fanculo Dolce & Gabbana non metto la roba fr*cia¹⁰

Moreover, many non-homotransphobic tweets actually share some linguistic similarities with the homotransphobic ones:

DI ANORMALE c'è solo che una cripto ch*cca repressa e omofoba quale lei è #Pillon sia miserabile Senatore della Repubblica pagato dagli italiani e che peggio getta discredito sulla nostra Nazione con esternazioni quotidiane di puro, spregevole letame.¹¹

Here, some hateful content is reported towards a person that is considered homotransphobic. In those cases, the presence of hate speech is detected correctly, but it does not meet the homotransphobic requirement.

¹⁰Fuck Dolce & Gabbana I'm not wearing that f*g stuff

¹¹ABNORMAL there is only that a repressed and homophobic crypto queer like you #Pillon is a miserable Senator of the Republic paid by the Italians and worse, discredits our nation with daily utterances of pure, despicable manure.

3.4 Discussion

The works reported in this chapter, which share the usage of relative light-weight models and encoder-only LMs, represents the early stages of the presented research work in NLP field. The proposed approaches stressed the capabilities of LMs in embedding extraction, combined with DNNs, such as LSTM, BiLSTM, transformers, and dense layers, for RE (SemEval 2010 Task 8) and classification tasks (HaSpeeDe3 and HODI). Diverse models have been tested and an overview of the experiments conducted is reported in Table 3.14.

Table 3.14: Overview of the conducted experiments in Chapter 3.

Experiment description	References
Testing of the proposed encoding strategies and architectures with SemEval 2010 Task 8	Data set overview in Table 3.1. Proposed architecture in Figures 3.2 and 3.3. Experimental results in Tables 3.2 and 3.3.
Testing of the proposed architectures with HaSpeeDe3 challenge	Data set overview in Tables 3.4, 3.5 and 3.6. Proposed architecture in Figure 3.4. Experimental results in Tables 3.7 and 3.8.
Testing of the proposed architectures with HODI challenge	Data set overview in Tables 3.9, 3.10 and 3.11. Proposed architecture in Figure 3.5. Experimental results in Tables 3.12 and 3.13.

The results obtained, even though they are not as enthusiastic compared to current SOTA systems, represent the starting point for all subsequent workflows, along with a parallel study of the models and architectures and the data set used themselves. A fine-tuning step is necessary for networks with encoder-only models to obtain significant results. Even in terms of computational costs, for the considered RE and classification task, a 12GB GPU was sufficient for both train and inference.

For future work, methodologies involving external Knowledge Graph (KG), in the context of document-level RE as DocRED [146] and E3C Corpus [84], have been taken into account. Meanwhile, the spread of newer models, such as the decoder-only LLMs and their application and promising performances in traditional NLP tasks [48] that started gaining popularity in the scientific community, influenced the subsequent research phase in generative approaches in text-only and multimodal models.

Chapter 4

Unipa-GPT framework

In this chapter are presented the works related to the Unipa-GPT framework [123, 125, 124, 126].

Unipa-GPT is a virtual assistant of the University of Palermo (Unipa) that helps secondary school students to explore the huge information provided by the University institutional website, when they are choosing a bachelor degree course, answering questions related to the available courses, application procedure, tax payments, access to scholarships, and more. Unipa-GPT was developed and presented to the broad public in the context of the SHARPER Night event¹ in 2023 and 2024.

More in detail, the following RQ will be dealt in this Chapter:

RQ 2 How can we use a decoder-only model to develop a chatbot and a QA application with domain-based IR for educational domain?

A RAG-based architecture was developed leveraging both closed- and open-source models, such as models from the GPT and Llama families, used as generative models. In addition, a documentary corpus, called Unipa-corpus, was built using a scraping procedure on the website of the University of Palermo (Section 4.2). Leveraging the RAG approach, relevant document to the given query of a user, are selected and used as a context to generate an answer. The generation procedure involved a fine-tuning procedure, but also the prompt design and generative parameters for the models. The evaluation of the developed models was partially performed through a qualitative evaluation based on feedback collected from people who used the chatbot during SHARPER Night in 2023 and 2024 (Sections 4.3).

¹<https://www.sharper-night.it/>

To overcome the previous limitations in the evaluation phase, the UniQA data set was built. It is a balanced English and Italian QA data set created from an updated version of Unipa-corpus, collecting documents natively written in Italian and English, from which nearly 14k QA pairs have been generated in a semi-automatic manner. Thanks to the availability of golden answer, generative capabilities of selected decoder-only model, queried with a proper prompt, a question and the relevant context, can be properly trained and evaluated (Section 4.4).

4.1 Related works

The scientific community’s interest in chatbots has started in the 1960s, when ELIZA [144] was developed, and, in general, chatbos applications can be seen as a QA task with memory. In particular, QA can be treated as a closed task, from now on it will be referred as Closed Question Answering (CQA), in which, given a question and a context, the answer is already reported in the context, and the task is to determine the relevant spans in the context and output them as an answer. On the other hand, what may be desirable in interactive setups, i.e. when dealing with chatbots and in real-world applications, is a system that freely answers unexpected questions for which a golden answer cannot be defined in advance since the answer is not known before, thus operating in an open setup, which hereafter will be referred to as Open Question Answering (OQA).

While CQA is a suitable task for encoder-only systems, challenges proposed in the OQA are deeply suitable for generative decoder-only LLMs, such as GPT family models [14, 97] on Llama ones [133, 82]. Specifically, GPT models became more popular after their use in ChatGPT, whose extraordinary generation capabilities overcome preexisting models, and a new SOTA level was redefined, and both researchers and people of interest were reached.

Generally speaking, every traditional NLP tasks can be proposed as a generative one, e.g. as a question to which a model has to answer. RAG [70] approach can be used in OQA context: generative capabilities of a given LLM are enhanced with the injection of external knowledge, thus effectively guiding the generation towards a correct answer. This approach can be effectively used in chatbots, since it relies on the assumption that the answer to user’s unexpected query can be derived from a well-selected context, according to the given query. In this setup, the role of the LLM is to generate a coherent answer to the question, while extracting relevant information from the provided context and re-paraphrasing them accordingly. Moreover, on doing this, the generation capabilities of the model are limited, thus resulting in reduction of incorrect answers and hallucinations in the generated answers.

In recent literature, different applications can be found as a helpful domain-based chatbot using a RAG approach, such as chatbots for Frequently Asked Questions (FAQ) [63], for information about the restaurant industry [11], assistance to victims of sexual harassment [134] or to the patient in a healthcare context [138, 139]. The education domain is also an interesting domain for these applications, such as improving user interaction for educational and research purposes [34], or serving as assistants for instructors [83], such as in generating questionnaires [112], exploring the pedagogical abilities of conversational models [130], or helping students with interactive learning approaches that can be personalized for their specific needs [110], such as Specific Learning Disorders (SLDs) [30].

Despite the target domain, all the aforementioned applications share methodologies similar to those of RAG approach, consisting in collecting a corpus of documents for the domain of interest and from which the answers to the users' questions can be derived. The collected documents are then divided into chunks and stored in a suitable data structure, such as a vector database, after the corresponding embeddings are calculated via a suitable LLM. At inference time, given a user question, the relevant chunks are retrieved from the vector database, leveraging a similarity distance, such as the cosine similarity. Retrieved chunks and user question, combined with a designed prompt, serve as input to a generative LLM that outputs an answer.

Although there is increasing interest in the research community, only a few applications involve the usage of Italian language, such as the helpful chatbots assistant for Italian Funding Application [12], or in the medical domain [39] or in the industrial context [33]. Those architectures use GPT-based models, both for generation and evaluation purposes. In this Chapter the analysis will focus on GPT-based models, but it will also be extended to open-source multilingual models [82, 98] and Italian adaptation of existing English models [5, 115, 8, 105].

In Section 4.2 the building process of an Italian document corpus will be covered, while in Section 4.3, OQA systems with both GPT and open-source models will be presented in the educational context of Unipa-GPT.

As for CQA, annotated QA data sets and specifically the Italian ones are not so common. A valuable example is SQuAD-it [24], derived by the English QA data set SQuAD [111], which collects more than 60k QA pairs obtained via a semi-automatic translation procedure. Generally speaking, data sets obtained via translations can be useful when a large amount of native data in Italian are not available, but they do not have the quality of manually annotated ones. On the other hand, QUANDHO [89] represents a closed-domain QA data set built from native Italian texts that collects 627 manually classified questions, thus reaching a high level of data quality, but its size is moderate.

In Section 4.4, an attempt is made to fill this gap, by the process of creation of a new QA data set with a considerably large number of manually generated prompts for questions and answers, which are based on structured data in both Italian and English without using any translation procedure.

4.2 Unipa-corpus

In this Section, the building process of the Unipa-corpus² is reported. In particular, two different versions were built; the former is a collection of Italian documents obtained from the University’s institutional website, used as a retrieval corpus for the developed RAG system (Section 4.2.1), the latter is an automatically generated fine-tuning corpus to properly train an LLM with an instruction-tuning strategy.

4.2.1 Unipa-corpus for RAG

The document corpus used for Unipa-GPT, called Unipa-corpus, is a collection of documents that were gathered directly from the website of the University of Palermo during Summer 2023. A manual selection of the most interesting web-pages was made, with reference to the target audience of secondary-school students. In particular, two main sections can be identified in the corpus, corresponding to the *Education* and the *Future Students* sub-trees in the institutional website. Since the expected questions are in Italian, the generated corpus is in Italian.

The *Education* section is the main part of the corpus and is a collection of all the courses available at the University for the academic year 2023/2024. For each course and each curriculum two documents are obtained: *details* is the document that collects all the general details of the course, such as name, department of affiliation, typology of course (Bachelor or Master degree), restriction of access and a colloquial description of the course, including its educational objectives and professional opportunities. *Course outline* is the second document that collects the course outline divided by year, and the number of credits, the teaching professor, the teaching period, and the scientific sector are specified for each class.

Three different versions of the course outline document were generated, namely clear, full, and emb. The *clear* version is the one described above, and the *full* version adds a new document, namely *class details*, for every class in a course and reports its peculiar educational objectives. The *emb* version is a mix of the previous ones in which the educational objectives of the classes are added directly to the document that contains the outline of the course. Despite this corpus is called as an abbreviation of embedded, it does not contain embeddings: the term embedded refers to the educational objectives that are inserted in the document containing the course outline. As a consequence, both the clear and the emb corpus have the same number of documents but different information, while the full contains the same information of the emb corpus, but it is arranged in a different number of

²<https://github.com/CHILab1/UnipaGPT-23>

details document	clear version of the course outline document	emb version of the course outline document	class details document in the full version
Anno Accademico 2023/2024 Laurea magistrale in INGEGNERIA INFORMATICA Curriculum: INTELLIGENZA ARTIFICIALE ----- Classe di Laurea magistrale in Ingegneria informatica (LM-32) Durata 2 anni Sede PALERMO Accesso libero Dipartimento di Ingegneria Doppio titolo con UNIVERSITE D'ARTOIS, Arras (FRANCE) Codice corso 2035 ----- Obiettivi Formativi Obiettivi specifici: Gli obiettivi formativi della Laurea Magistrale [...] ----- Sbocchi Occupazionali Profilo: Ingegnere informatico magistrale [...] Funzioni: Progettista/amministratore di sistemi informatici [...] Competenze: Incremento della conoscenza [...] Sbocchi: La libera professione [...] ----- Caratteristiche della Prova Finale Per il conseguimento [...] -----	Anno Accademico 2023/2024 Laurea magistrale in INGEGNERIA INFORMATICA Curriculum: INTELLIGENZA ARTIFICIALE Per maggiori informazioni vai su http://www.unipa.it/struttura.html?id=721 ----- Classe di Laurea magistrale in Ingegneria informatica (LM-32) Durata 2 anni Sede PALERMO Accesso libero Dipartimento di Ingegneria Doppio titolo con UNIVERSITE D'ARTOIS, Arras (FRANCE) Codice corso 2035 ----- Insegnamenti primo anno Codice materiaNome materia DocenteCFU PeriodoSSD Obiettivi Formativi 03994INTELLIGENZA ARTIFICIALE I GAGLIO9 1° semestreING-INF/05	Anno Accademico 2023/2024 Laurea magistrale in INGEGNERIA INFORMATICA Curriculum: INTELLIGENZA ARTIFICIALE Per maggiori informazioni vai su http://www.unipa.it/struttura.html?id=721 ----- Classe di Laurea magistrale in Ingegneria informatica (LM-32) Durata 2 anni Sede PALERMO Accesso libero Dipartimento di Ingegneria Doppio titolo con UNIVERSITE D'ARTOIS, Arras (FRANCE) Codice corso 2035 ----- Insegnamenti primo anno Codice materiaNome materia DocenteCFU PeriodoSSD Obiettivi Formativi 03994INTELLIGENZA ARTIFICIALE I GAGLIO9 1° semestreING-INF/05 Il corso si propone di fornire allo studente i concetti di base nell'ambito dell'intelligenza artificiale. Durante il corso vengono affrontate le tecniche di problem solving, di ragionamento automatico e di apprendimento automatico, assieme alla loro applicazione a casi concreti.	Il corso di INTELLIGENZA ARTIFICIALE 1 del corso di Laurea magistrale in INGEGNERIA INFORMATICA del curriculum INTELLIGENZA ARTIFICIALE, ha i seguenti obiettivi formativi: Il corso si propone di fornire allo studente i concetti di base nell'ambito dell'intelligenza artificiale. Durante il corso vengono affrontate le tecniche di problem solving, di ragionamento automatico e di apprendimento automatico, assieme alla loro applicazione a casi concreti.

Figure 4.1: Overview of the details and course outline and documents for each corpora version.

documents. A simplified overview of the built documents is reported in Figure 4.1.

The *Future Student* section is the same for the three versions of the corpus and is a mixture of documents coming from the related section of the University website. The information contained in these documents is directed to future students of the University, including the academic calendar, the tax rules and reductions, scholarships, University enrollment procedure, and facilities offered to the students. The statistics of each corpus are reported in Table 4.1.

Table 4.1: Number of documents in each version of Unipa-corpus.

	Education	Future Students	Total documents	Total tokens
Unipa-corpus-clear	506	104	610	2.059.638
Unipa-corpus-full	5.794	104	5.898	890.598
Unipa-corpus-emb	506	104	610	2.063.020

4.2.2 Unipa-corpus for fine-tuning

The Unipa-corpus was modified to be in the form required for fine-tuning a generative LLM, such as a triple made up of a prompt, an input and an output. In the prompt, the instruction LLM is expected to follow, the input is the actual user query, and the instruction is the desired correct answer. Since the idea is to leverage the API for fine-tuning GPT-3.5-turbo, which can be accessed by payment, the main intent is lowering as much as possible the computational resources, thus allowing an on-budget fine-tuning. This results in using the minimum tokens to train the model. In addition to the economic aspect, in the case of GPT fine-tuning, this is a crucial topic when dealing with LLMs because also relatively small LLMs like LLama 2 in the 7B version [133] require huge computational resources for their full fine-tuning or retraining.

To overcome this problem, PEFT techniques as LoRA [49] have been developed, to reduce the required computational resources for a satisfactory LLM fine-tuning, but it is not available for closed systems such as GPT-3.5-turbo. Since the primary purpose is towards stressing the capabilities of ChatGPT, the performances of the model were also tested after a fine-tuning phase over the Unipa-corpus-clear, the smallest created corpus in terms of tokens to be trained thus reaching a trade-off in terms of API costs.

The required format for fine-tuning is a sequence of prompt, question and answer, as below.

Prompt

Sei Unipa-GPT, un assistente virtuale che risponde alle domande sull'università di Palermo.

Question

Parlami del corso di Laurea magistrale in INGEGNERIA INFORMATICA curriculum INTELLIGENZA ARTIFICIALE

Answer

Il corso di Laurea magistrale in Ingegneria Informatica curriculum Intelligenza Artificiale ha una durata di 2 anni ed è accessibile liberamente. Si svolge presso il Dipartimento di Ingegneria a Palermo e prevede un doppio titolo con l'Università d'Artois [...]

No additional context (e.g. the closest documents for an effective reply of the model) was added to the prompt for fine-tuning, thus maintaining low the costs of

the whole process. The prompt used in this context is a simple instruction on chat-bot behavior. Questions and answer were automatically generated with different generation strategies for each different section of the corpus. For documents in the Education section, corresponding QA pairs were automatically generated asking GPT-3.5-turbo to answer the provided question with the information contained in the document provided as a context, as well as for both the course details and the outline document, as in Table 4.2.

Table 4.2: Generation prompts for QA pairs for the Education section

Course details	Course outline
Rispondi alla domanda usando le informazioni contenute del documento. Documento: COURSE DETAILS DOCUMENT Domanda: Descrivi il corso di NOME CORSO	Rispondi alla domanda usando le informazioni contenute del documento. Documento: COURSE OUTLINE DOCUMENT Domanda: Cosa si studia nel corso di NOME CORSO?

The LLM was asked to provide the topics of a specific degree course starting from the corresponding course outline document. In both cases, the golden document was given along with the question, and the given answer was considered an answer to the fine-tuning corpus. As regards the Future Students section, QA pairs were extracted directly from the documents that already contain a FAQ section, otherwise the remaining pairs were generated manually. In this case, a directed question related to a document section was formulated and the answer was extracted from the document. Alternatively, a generic request was formulated like *parlami di ...*³. Some documents were not considered in their entirety since the information contained was highly specific and related to non-relevant topics.

Table 4.3: Number of documents in the Unipa-corpus-clear re-arranged for fine-tuning.

	Education	Future Students	Total QA pairs	Total tokens
Training set	506	269	775	259.772
Validation set	253	133	386	123.118

A validation set was also expunged from the training data by both sampling and re-paraphrasing the generated questions. A QA pair was randomly picked

³speak about ...

for each degree course among the details and outline documents of the course in the Education section. The statistics of the corpus for fine-tuning are reported in Table 4.3. Despite this arrangement can be suitable for creating also a test split of the data set, to further evaluate fine-tuning performances of the model, the primary focus in this phase was towards determining the models that better matches users' preferences. Moreover, the principal objective of fine-tuning was to inject domain-specific knowledge into the model, thus allowing it to generate more precise answers when queried.

GPT-3.5-turbo was the only powerful model available at the time of the experiments⁴ to automatically generate QA pairs. As can be inferred from Table 4.3, manual generation of the total number of QA pairs would be impractical. It is also clear that possible biases can arise in fine-tuning a model with self-generated samples. The so generated QA pairs, served as fine-tuning data set, whose scope was to achieve in the fine-tuned model, a terminological knowledge of the domain through the presentation of a small subset of the facts contained in the Unipa-corpus. With this consideration, the risks associated with the introduction of any relevant bias are mitigated, also because, at inference time, the model is inserted in a RAG architecture that is queried using an instruction prompt strategy.

⁴July-September 2023

4.3 Unipa-GPT

Unipa-GPT⁵ is a virtual assistant capable of answering to secondary school students who need information for accessing and studying at the University of Palermo, which is built upon a RAG architecture [70], made up of two main components, a retriever and a generator module. Two main versions were developed, the former, simply referred to as Closed Unipa-GPT (Section 4.3.1), that leverages on OpenAI models, while Open Unipa-GPT (Section 4.3.2), open-source models were involved.

4.3.1 Closed Unipa-GPT

In this Section, the architecture of closed Unipa-GPT is presented (Subsection 4.3.1.1) along with the preliminary experimental results and the experience during the SHARPER nigh, during which the model was presented and used by real users (Subsection 4.3.1.2).

4.3.1.1 Architecture

Closed Unipa-GPT is developed as a RAG architecture [70], made up of two main components, as shown in Figure 4.2: the retriever and the generator module.

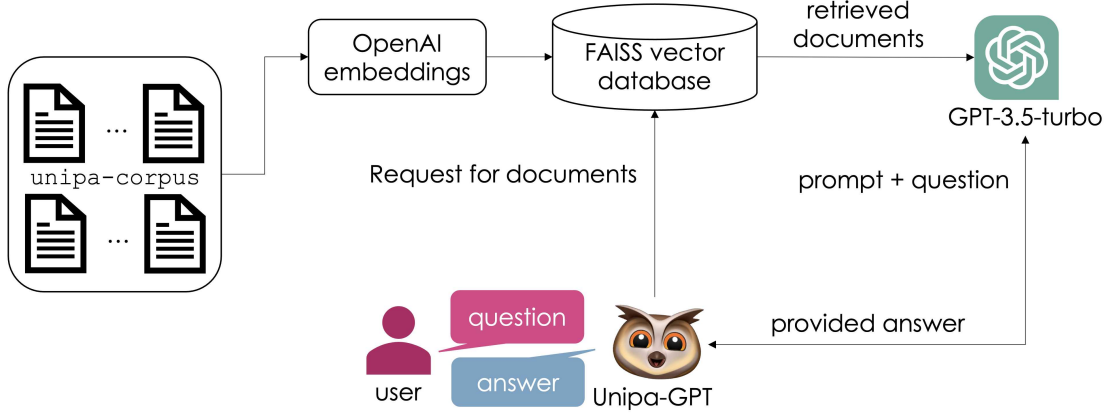


Figure 4.2: The architecture of the RAG version of Closed Unipa-GPT.

⁵<https://github.com/CHILab1/UnipaGPT-23>

The *retriever module* consists of a FAISS [29] vector database, filled with the documents in Unipa-corpus conveniently divided into chunks of 1000 tokens with an overlap of 50 tokens. Embeddings were extracted using text-embedding-ada-002 by OpenAI⁶ as Embeddings LLM.

The *generator module* consists of an instance of GPT-3.5-turbo [14], which serves as Generator LLM. The model is queried with a custom instruction in which the behavior of the system is explained, and the input is formed by the question of the user combined with the most related documents. The expected answer of the system is a reply to the question of the user according to the instruction that rules the Unipa-GPT overall behavior and the domain-knowledge given by the retriever. GPT-3.5-turbo makes inferences using a temperature hyperparameter equal to 0 thus its behavior is as deterministic as possible, and the system is prevented from being creative, thus limiting the output of erroneous answers. Finally, no limits *a priori* were placed on the maximum tokens available for the answer, to prevent broken answers. The chatbot behavior and the overall RAG architecture was implemented via the LangChain library⁷ that allows both the retrieval and inference interface, allows the chat history to be kept and the ChatGPT behavior to be simulated through the GPT-3.5-turbo API.

Table 4.4: Overview of the instructions used in the Closed Unipa-GPT.

instruction type	instruction text
<i>custom instruction</i>	Sei unipa-gpt, il chatbot e assistente virtuale dell'Università degli Studi di Palermo. Rispondi cordialmente e in forma colloquiale alle domande che ti vengono poste. Se ricevi un saluto, rispondi salutando e presentandoti. Se ricevi una domanda riguardante l'università degli studi di Palermo, rispondi in base ai documenti che ti vengono dati insieme alla domanda. Se non sai rispondere, scusati e suggerisci di consultare il sito web, non inventare risposte. Question: {question} Documenti: {context}
<i>condensed instruction</i>	Data la seguente conversazione e la domanda successiva, riformula la domanda successiva in modo tale sia una domanda singola. Conversazione: {chat_history} Domanda successiva: {question} Domanda singola:

Both GPT-3.5-turbo and text-embedding-ada-002 were invoked via Azure call to the OpenAI API. Two different Italian instructions were built for GPT-3.5-turbo: a custom instruction and a condensed instruction, as shown in Table 4.4

⁶<https://openai.com/blog/new-and-improved-embedding-model>

⁷https://python.langchain.com/docs/get_started/introduction

(English translation can be found in Table 7.1 in Appendix). The *custom instruction* is an explanation of the behavior of the chatbot where both the previous conversation and the new question are concatenated to the prompt itself. In contrast, *condensed instruction* adds another instruction to the custom instruction to condense the previous conversation and re-arrange it as a single new question that will be answered accordingly to the custom instruction.

In the construction phase of custom instruction, it was tested in a preliminary manner with simple questions such as *What do you study in computer engineering?* or *How do I enroll at the university?*. Here the focus was towards explicit in the textual instruction how the model should answer, the tone that should be used. In addition, the model was prompted to avoid answering out-of-context questions, to rely only on the provided documents, which serves as context to generate the final answer, and to prevent answering if the provided documents do not have the required information. In the last case, the idea was to prevent the model from generating potential incorrect answers, which may lead to the spread of false information.

The idea in prompt construction was towards an in-context learning approach [14], in which neither CoT [143] nor many-shot prompting strategies have been used. Since the core of Unipa-GPT is to provide information, force the model to provide insights and explanation on the generated answer, is not necessary nor useful to the end-user. In addition, a few-shot prompting strategy will result in a longer prompt in which, in addition to the target question and the associated relevant documents, also samples made up of a question, relevant documents for the given question and the annotated answer should be provided. In the context of closed model, which usage undergoes with paid API, the choice was towards a simple and conservative prompt strategy.

In addition to the RAG version illustrated above, a fine-tuned version was implemented with the custom fine-tuned version of GPT-3.5-turbo where the Unipa-corpus explicitly re-arranged, as described in Section 4.2.2, was used. The same RAG architecture and instruction instances mentioned above were used in the fine-tuned model. The whole system is shown in Figure 4.3.

4.3.1.2 Results

This Section reports the experimental results obtained in three different scenarios: Subsection 4.3.1.2 illustrates the preliminary development phase, which took place during the summer of 2023. In Subsection 4.3.1.2 is reported the analysis over the results obtained during the SHARPER Night 2023 and 2024, where the best resulting model from the previous phase was used in free interaction with the public. In these two scenarios, a qualitative metric based on human judgment was used for evaluation purposes. In this phase, the evaluation was more geared

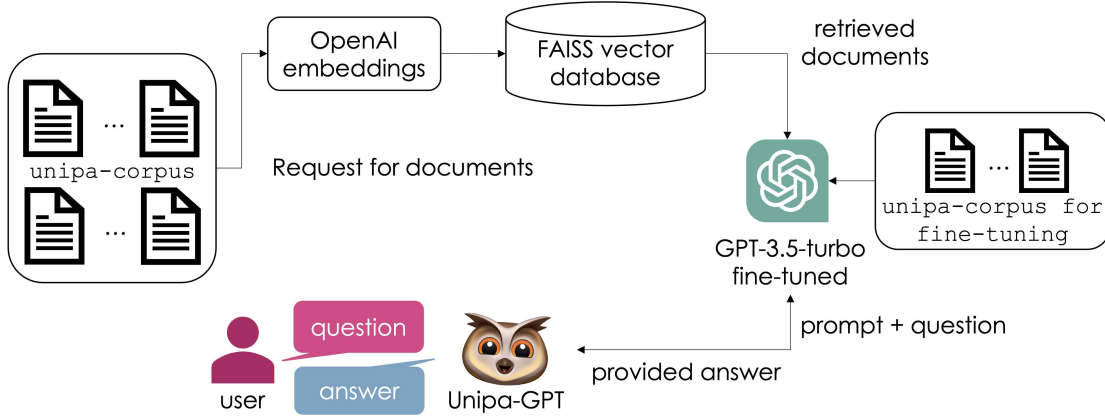


Figure 4.3: The schema illustrates the architecture of the proposed system where GPT-3.5-turbo-fine-tuned is used after a fine-tuning process over Unipa-corpus.

towards investigating the preferences of the users regarding the feedback provided by the chatbot during the interaction.

Preliminary experiments

This Section reports the results of the very early evaluation runs performed using both versions of Closed Unipa-GPT, namely the RAG-only and the fine-tuned one, before deploying the system to the public during the SHARPER Night 2023. These first experiments aimed to gain at least qualitative information on the structure of the prompt to be used to query the system. The objective was to assess the best performing Closed Unipa-GPT version, provided that the size of the data set was very small compared to the training data used for GPT-3.5-turbo.

The idea behind Unipa-GPT relies on building a chatbot designed to answer questions posed by secondary school students, and, during an Open Day activity at the University of Palermo, these students were asked to interact with demo versions of the system. Starting from real doubts, two lists of Italian questions were devised, and students were asked to chat with the demo version of Closed Unipa-GPT. Therefore, they were asked to express a qualitative judgment of the responses received, thus evaluating the closed performance of Unipa-GPT. Both lists of questions, called Chat1 and Chat2, address similar topics and have been written by students in an unsupervised environment in which they only knew that they are interacting with an AI system able to answer on university-related questions. This unsupervised scenario was built to simulate a real scenario in which the user has no external guidelines towards a more correct or incorrect way to pose questions.

Chat1, consists of 14 questions about the subscription procedure, courses in the Department of Engineering, suggestions related to the Bachelor course to apply for in order to access a specific Master course, information about University admission, test, and questions about classes in the first year. Conversely, Chat2 contains 10 questions with typographical errors and grammatical errors. The questions in Chat2 are expressed more like a Google search or a list of keywords. The topics involved in Chat2 are related to taxes, Erasmus programs, scholarships, and the Student Desk.

Table 4.5: Evaluation results of the different runs for both Chat1 and Chat2 over Closed Unipa-GPT.

Custom instruction				Custom instruction FT			
	<i>excellent</i>	<i>good</i>	<i>bad</i>		<i>excellent</i>	<i>good</i>	<i>bad</i>
Chat1-emb	6	1	7	Chat1-emb	4	6	4
Chat1-full	6	4	4	Chat1-full	4	4	6
Chat1-clear	8	4	2	Chat1-clear	6	4	4
Chat2-emb	4	2	4	Chat2-emb	5	1	4
Chat2-full	3	3	4	Chat2-full	6	3	1
Chat2-clear	8	1	1	Chat2-clear	6	1	3

Condensed instruction				Condensed instruction FT			
	<i>excellent</i>	<i>good</i>	<i>bad</i>		<i>excellent</i>	<i>good</i>	<i>bad</i>
Chat1-emb	5	1	8	Chat1-emb	5	2	7
Chat1-full	10	2	2	Chat1-full	4	1	10
Chat1-clear	10	1	3	Chat1-clear	6	0	8
Chat2-emb	6	2	2	Chat2-emb	5	1	4
Chat2-full	8	2	0	Chat2-full	6	1	3
Chat2-clear	9	0	1	Chat2-clear	8	0	2

Both Chat1 and Chat2 were issued using different closed Unipa-GPT configurations with respect to the corpus used in the retriever (clear, full, emb) the instruction (custom instruction, condensed instruction) and the LLM training procedure (GPT-3.5-turbo, GPT-3.5-turbo-fine-tuned) for a total of 12 runs. Different Closed Unipa-GPT configurations are referred to as the instruction name used, and, if a

fine-tuned version of GPT is used, it is marked as FT. Each student was asked to simply classify each answer as “excellent”, “good” and “bad”, allowing for easier and intuitive evaluation, from real end users, instead of a more traditional linguistic evaluation, for which a golden answer is needed. Answers with a “*excellent*” label are the best ones, while answers with a “*good*” label are not optimal and precise, but contain a partially good answer and may add non-requested bits of information that are less related to the initial question. “*Bad*” answers are either answers that provide incorrect information or replies off-topic along with any unexpected or unwanted behavior of the system. In addition, replies to questions that the system cannot answer are considered bad. Both Chat1 and Chat2 contain questions that are strictly related to previous answers from the system, thus simulating the case when the user asks for further information. As an example, if the user asks for the admission test to a particular Bachelor course, and the system replies that the course is a free admission one, the (expected) question regarding test rules is not applicable, thus it is not posed to the system, and the missed reply is labeled as bad answer. Table 4.5 shows the evaluation results.

The two best-performing models use both the condensed instruction and GPT-3.5-turbo, and Unipa-corpus-full and Unipa-corpus-clear, respectively, as for the retrieval corpus. The model that uses the clear corpus was selected because it is the one with the highest number of excellent labels independently of the prompt (18 labels in total). The lack of additional information on the educational objectives of the classes in the retrieval corpus makes it more precise. However, the use of the condensed instruction produces a less degrading conversation compared to the use of the custom instruction, where the conversation is passed as it is, without refinements. On the other hand, it is worth noting that none of the models performs poorly during the entire conversation.

No performance improvements were found in the fine-tuned models, revealing that this strategy is not effective for the target task, and a performance decrease is observed in the configurations with condensed instruction. Generally speaking, the answers of the fine-tuned models are very long and tend to be repetitive [47] and to generate loops and hallucinations [52, 152]. More in detail, fine-tuned models enrich their answers also with extra pieces of information that sometimes are pertinent, e.g. when the model is asked for information regarding taxes payment, the fine-tuned model, in addition to general information as the non-fine-tuned one, suggests the possibility for payments reduction as reported in Table 4.6 (English version in Table 7.2 in Appendix). In other cases, fine-tuning leads to a degradation in the quality of answers, with both out of topic replies or contradictions, as shown in Table 4.7 (English version in Table 7.3 in Appendix). Here, a lack of coherence can be highlighted as previous information on the fact that mechanical

engineering is an open access course, but a test is required to enroll.

Table 4.6: Answers from Chat2 generated with condensed instruction from base and FT GPT and the Unipa-corpus-emb for the retrieval. Second answer is truncated due to its length: in the missing text other cases of taxes reduction are illustrated.

condensed-emb	condensed-FT-emb
<i>cos'è il contributo onnicomprensivo</i>	
Il contributo onnicomprensivo per l'immatricolazione all'Università degli Studi di Palermo per l'anno accademico 2023/24 è calcolato in base alla classe di reddito ISEE dello studente e alle prestazioni erogate nell'ambito del diritto allo studio universitario. Il pagamento delle tasse e del contributo onnicomprensivo è suddiviso in tre rate, la prima delle quali deve essere pagata entro il 30 settembre 2023 per l'immatricolazione ai corsi di studio ad accesso libero e per le iscrizioni ad anni successivi al primo. Per conoscere i termini e gli importi del pagamento delle tasse e del contributo onnicomprensivo, si consiglia di consultare il Calendario didattico e il Regolamento sulla contribuzione studentesca vigenti. In caso di dubbi o domande specifiche, si suggerisce di contattare direttamente la Segreteria Studenti dell'Università degli Studi di Palermo.	Il contributo onnicomprensivo è il pagamento delle tasse universitarie che gli studenti devono effettuare all'atto dell'immatricolazione o iscrizione. Il pagamento è suddiviso in tre rate: la prima rata, da versare all'atto dell'immatricolazione o iscrizione, nella misura del 40% dell'importo dovuto; la seconda rata, da versare entro il 29 dicembre 2023, nella misura del 30% dell'importo dovuto; la terza rata, da versare entro il 29 febbraio 2024, nella misura del 30% dell'importo dovuto. Gli studenti che si iscrivono per la prima volta alla Laurea triennale o Laurea Magistrale a Ciclo Unico, diplomatisi con voto compreso tra 95 e 100, usufruiscono, per il primo anno, di una riduzione del 25% fino ad un massimo di 350,00 euro. Coloro che si diplomano con voto pari a 100 e lode usufruiscono, per il primo anno, di una riduzione del 50% fino ad un massimo di 500,00 euro. [...]

With reference to the instructions, the condensed instruction outperforms, along with both the Unipa-corpus-full and the Unipa-corpus-clear corpus for the retrieval. Both corpora provide very pertinent and similar answers. In contrast, models with custom instruction tend to be less precise, as reported in Table 4.8 (English version in Table 7.4 in Appendix). In this case, the answer generated by the RAG-only model is more accurate since the advice provided to enroll in the bachelor course in Mechanical Engineering with Aeronautical curriculum is more precise and correct than the generic suggestion towards Mechanical Engineering. In addition, the first answer is also the same as in the context of Open Day activities that will be provided to interested future students.

All models share the same negative aspects: all the provided answers are well written in Italian, and an uninformed user can easily consider a wrong answer as true by looking at both the lexicon and the grammar used by the system. Sometimes, the prompt structure causes the system to answer with an out-of-context presentation of Closed Unipa-GPT, and a repetition of the question is necessary to obtain the expected response. In this situation, prompt tuning plays a key role in achieving the best performance [14, 157]. Discussed and reported QA pairs, refer to the experiments in Table 4.5, for which only the most significant highlights are reported.

Table 4.7: Answers from Chat1 generated with custom instruction: the fine-tuned model contradicts itself.

custom-full	custom-FT-clear
<i>la triennale in ingegneria meccanica è a numero chiuso o a numero aperto?</i>	
La triennale in ingegneria meccanica presso l'Università degli Studi di Palermo è a numero aperto, con un numero di posti in programmazione locale pari a 200. Questo è indicato nel documento relativo all'anno accademico 2023/2024 per il curriculum meccanico. Se hai bisogno di ulteriori informazioni, ti suggerisco di consultare il sito web dell'università	La triennale in ingegneria meccanica è a numero aperto.
<i>devo fare test per entrare alla triennale in ingegneria meccanica?</i>	
Sì, per accedere alla triennale in ingegneria meccanica all'Università degli Studi di Palermo è necessario sostenere un test di accesso. Il Centro Orientamento e Tutorato (COT) dell'Università organizza diverse iniziative per affrontare al meglio i test di accesso e superarli con successo, come i workshop "Come affrontare il test di accesso" e la simulazione dei test di accesso. Inoltre, sono disponibili corsi di preparazione alle prove di accesso che inizieranno il 17 luglio 2023 e si svolgeranno esclusivamente online sulla piattaforma Microsoft Teams. Per maggiori informazioni, ti consiglio di consultare i documenti forniti insieme alla domanda.	Sì, sono previsti test per entrare alla triennale in ingegneria Meccanica.

Table 4.8: Answers from Chat1 generated leveraging Unipa-corpus-clear as the retrieval corpus. Both the answers are correct but the one reported by the RAG-only model is more accurate.

condensed-clear	custom-FT-clear
<i>quale corso triennale mi consigli di frequentare per poi passare alla magistrale di aerospaziale?</i>	
Per passare alla laurea magistrale in Ingegneria Aerospaziale all'Università degli Studi di Palermo, il corso triennale consigliato è la Laurea in Ingegneria Meccanica con curriculum Aeronautico. Ti consiglio di consultare il piano di studi del corso di laurea triennale e di contattare il Dipartimento di Ingegneria per ulteriori informazioni.	Ti consiglio di frequentare il corso di Laurea triennale in Ingegneria Meccanica per poi passare alla magistrale in Ingegneria Aerospaziale. Il corso triennale ti fornirà le basi necessarie per affrontare al meglio il percorso magistrale.

Closed Unipa-GPT performs in free QA domain where the user can be anyone with different background, thus questions submitted to the system are deeply variable; the developed models result in more sensible results and perform better or worse depending on the specific question. In Chat2, questions about scholarships and classes were made as a request and not as a question. This led the system

either to answer badly or not to provide an answer at all by making its excuses, as reported in Table 4.9 (English version in Table 7.5 in Appendix).

In Table 4.9 the behavior of the fine-tuned version of GPT-3.5-turbo is also reported with the same instruction used in the training phase and without the access to Unipa-corpus, to test the effectiveness of fine-tuning: the fine-tuned model provides an answer to both the proposed questions (even to the no-question), but both are wrong, despite they are credible.

Table 4.9: Answers from Chat2 generated with base GPT-3.5-turbo and condensed instruction. Retrieval was performed, in the leftmost columns, with Unipa-corpus-emb and Unipa-corpus-full. The third model was queried only leveraging the fine-tuned version of GPT-3.5-turbo, with the same instruction used in training phase, i.e. without RAG.

condensed-emb	condensed-full	fine-tuned model without RAG
<i>lista esami primo anno di medicina</i>		
Gli esami del primo anno del corso di medicina all'Università degli Studi di Palermo per l'anno accademico 2023/24 non sono specificati nei documenti forniti. Ti suggerisco di consultare i bandi di concorso per maggiori informazioni. Inoltre, per quanto riguarda l'immatricolazione, i corsi di studio ad accesso libero possono essere immatricolati senza mora dall'1 agosto [...]	Mi dispiace, ma non ho trovato informazioni riguardanti gli esami del primo anno del corso di medicina all'Università degli Studi di Palermo per l'anno accademico 2023/24. Ti consiglio di consultare i bandi di concorso per maggiori informazioni.	La lista degli esami del primo anno del corso di laurea in Medicina e Chirurgia presso l'Università di Palermo prevede i seguenti esami: Anatomia Umana con Esercitazioni, Biologia Applicata, Lingua Inglese, Chimica Biologica con Elementi di Biologia Molecolare, Fisica, Matematica, Statistica e Informatica, Istologia Umana con Elementi di Biologia Cellulare, Fisiologia Umana con Esercitazioni, e Patologia Generale.
<i>quali sono le materie del primo anno di medicina</i>		
Le materie del primo anno del corso di medicina all'Università degli Studi di Palermo per l'anno accademico 2023/24 sono: Fisica I, Matematica I e Chimica Generale ed Organica C.I. Invece, per il corso di laurea in Scienze Biologiche, le materie del primo anno sono: Matematica con esercitazioni, Citologia e Istologia con esercitazioni, Chimica Generale con esercitazioni e Botanica Generale e istematica con esercitazioni.	Le materie del primo anno del corso di medicina all'Università degli Studi di Palermo per l'anno accademico 2023/24 sono: Fisica I, Matematica I, Chimica Generale ed Organica C.I., Chimica Organica, Biologia, Embriologia ed Istologia C.I., Genetica e Microbiologia C.I., Informatica, Bioinformatica e Inglese Tecnico Scientifico C.I. Ti consiglio di consultare i documenti forniti per maggiori informazioni.	Le materie del primo anno di medicina sono: Anatomia Umana, Istologia, Embriologia, Biochimica, Biologia Cellulare, Genetica, Fisica, Chimica, Lingua Inglese, Metodologia del Lavoro Intellettuale, e Cultura e Valori in Sanità.

SHARPER Night case study

Unipa-GPT was presented during the European Researchers Night (SHARPER Night), when visitors were allowed to interact with the chatbot making questions about the University of Palermo. The created interface, shown in Figure 4.4, was designed to be as user-friendly as possible with an avatar that mimics the answer of the system by moving itself and reproducing an audio version of the answer,

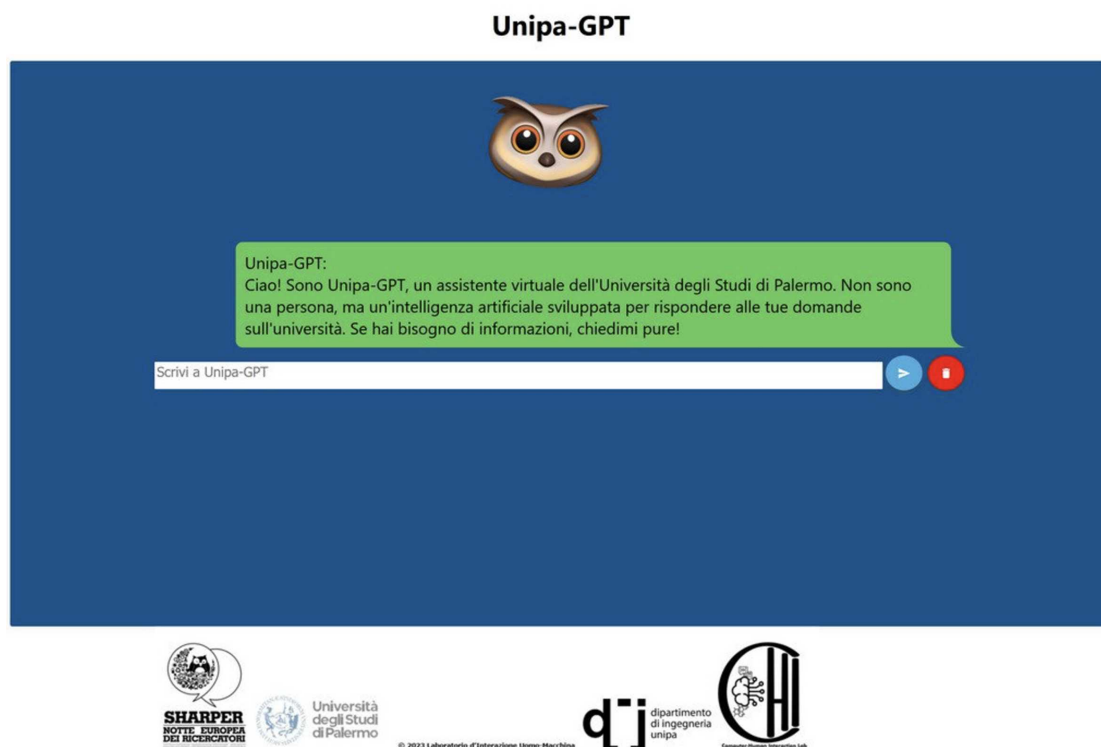


Figure 4.4: The GUI of Unipa-GPT presented at the SHARPER Night 2023.

making it more accessible and interactive. As shown in 4.4, the chatbot always starts a conversation displaying a welcome message in Italian.

Ciao! Sono Unipa-GPT, un assistente virtuale dell'Università degli Studi di Palermo. Non sono una persona, ma un'intelligenza artificiale sviluppata per rispondere alle tue domande sull'università. Se hai bisogno di informazioni, chiedimi pure!⁸

Some users were confident during the interaction, while others needed an external human guidance to interact with the chatbot, demonstrating that skeptical visitors about AI, in a real context, are less prone to give a chance to this type of system. The model that was actually used during the SHARPER Night uses the condensed instruction with GPT-3.5-turbo and Unipa-corpus-clear as the retrieval corpus: due to further experiments, few changes were made in the instruction, as shown in Table 4.10 (English version in Table 7.6 in Appendix). Those changes

⁸Hi! I am Unipa-GPT, a virtual assistant at the University of Palermo. I am not a person, but an artificial intelligence developed to answer your questions about the university. If you need any information, just ask me!

Table 4.10: The instructions used in the system during the SHARPER Night.

instruction type	instruction text
<i>custom instruction</i>	Sono Unipa-GPT, chatbot e assistente virtuale dell’Università degli Studi di Palermo che risponde cordialmente e in forma colloquiale. Ai saluti, rispondi salutando e presentandoti; Rispondi alla domanda con la dicitura “Risposta: ” Ricordati che il rettore dell’Università è il professore Massimo Midiri. Se la domanda riguarda l’università degli studi di Palermo, rispondi in base alle informazioni e riporta i link ad esse associate; Se non sai rispondere alla domanda, rispondi dicendo che sei un’intelligenza artificiale che ha ancora molto da imparare e suggerisci di andare su https://www.unipa.it/, non inventare risposte. Domanda: {question} Informazioni: {context}
<i>condensed instruction</i>	Data la seguente conversazione e la domanda successiva, riformula la domanda successiva in modo tale sia una domanda singola. Conversazione: {chat_history} Domanda successiva: {question} Domanda singola:

were required after the preliminary experiment to force the model to produce a better answer and provide a colloquial tone. As in the following textual frame *Se non sai rispondere alla domanda, rispondi dicendo che sei un’intelligenza artificiale che ha ancora molto da imparare*⁹, it was insert to provide a pleasant answer in the context of the SHARPER Night, while the precision of the name of the rector of the University was necessary since the cutoff date of GPT-3.5-turbo was prior to the designation of the new rector, and the model risked giving incorrect answers.

During the event, a total of 165 questions were collected and divided into seven categories, namely Generic Information, Course Information, Other University-related, Off-topic, Services and Structures, Taxes and Scholarships and University Environment (Figure 4.5.a). On average, an interaction consists of a two-questions chat, and a total of 31 feedback from visitors were collected via a Google Form.

The majority of the feedback collected came from university students or professors, that is, people who are familiar with the academic world (Figure 4.5.b) and the general rate of the chatbot is very positive, despite two bad rankings (Figure 4.5.c). A bad evaluation may come from a conversation in which a student

⁹if you do not know how to answer the question, answer by saying that you are an artificial intelligence that still has a lot to learn

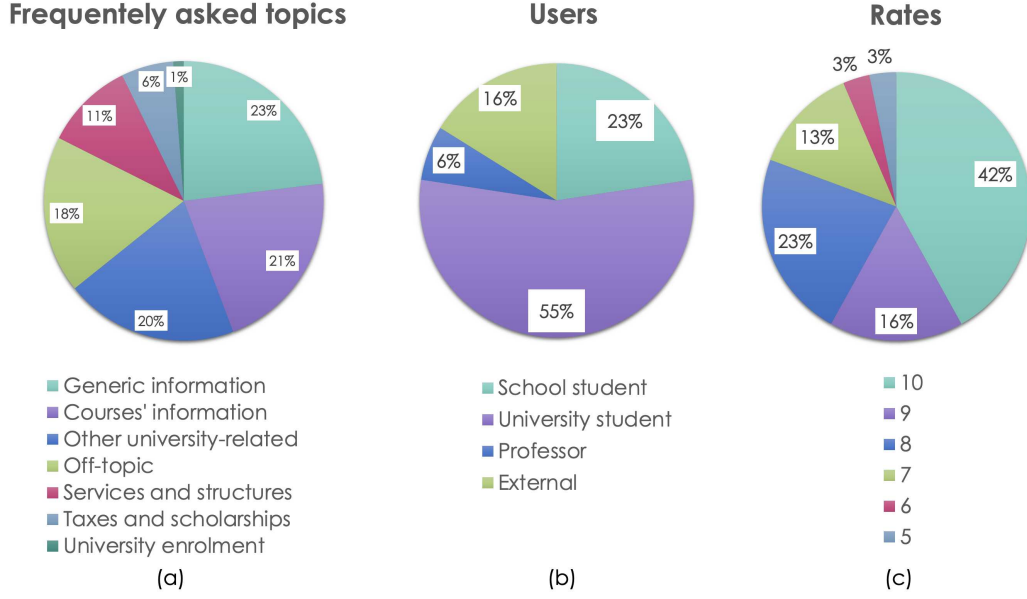


Figure 4.5: Charts illustrate the frequently asked topics to the model during the SHARPER Night 2023 (a), among the 31 feedback collected, an overview of the users is reported (b) and the rates to the application (c).

asked about the ISPE (Indicatore di Situazione Patrimoniale Equivalente¹⁰) value used to determine eligibility for a scholarship. In this particular case, the model provided a false answer, saying that the asked parameter is not considered for the rank, despite the truth. This can be considered as a case in which the model made use of an incomplete document. The second bad evaluation came from a conversation in which information about the final test for a bachelor's course was asked of the model. Here, the problem is the bachelor course name: the user did not use the complete official course name but the one to which it is usually referred to. Moreover, bachelor and master courses share the same name, and this may cause an ambiguity in the model that can be fixed by adding external information. In this case, the final test is peculiar to the bachelor's course, while the thesis refers to the master's course. It is reasonable to assume that if this kind of general knowledge is provided to the model, in addition to the document, the correct answer may be generated.

Unipa-GPT participated also in the 2024 edition of the SHARPER Night: instead of the GPT-3.5-turbo, the newer GPT-4o-mini was used, leveraging the same prompting strategy and instruction. The Unipa-corpus was updated with the

¹⁰Equivalent Asset Situation Indicator

courses available from the academic year 2024/2025 and with newer enrollment, taxes, and scholarship regulations. In this edition a total of 207 answers were collected for a total of 70 conversations with an average of 3 interactions, and they were manually classified as the aforementioned categories as in (Figure 4.6.a). In this edition, a considerable amount of questions falls into Off-topic section, mainly due to the interactions with school students that were highly fascinated in the usage of the application, but a domain-based interaction was difficult to reach. In contrast, the black box of GPT-4o-mini allowed for fluent conversions despite the topics being not university-related. In this edition, a total of 28 feedback was collected, mainly from university students and some students from primary school. In addition, in this evaluation, few low rates were assigned, mainly due to tricky questions to which the model had no access. In Unipa-corpus, as a design decision, no information related to the lecture schedule, details of how the final grade is calculated¹¹ are reported, nor the possibility to calculate the amount of taxes that had to be paid.

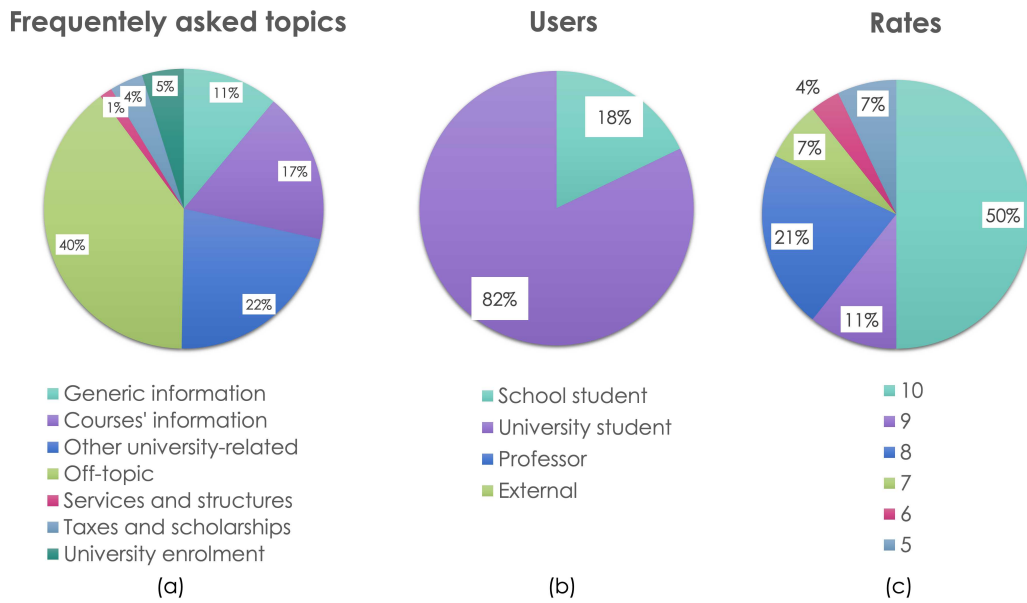


Figure 4.6: Charts illustrate the frequently asked topics to the model during the SHARPER Night 2024 (a), among the 31 feedback collected, an overview of the users is reported (b) and the rates to the application (c).

Generally speaking, the application meets the needs of the users and, despite some initial hesitation in interaction, the interaction experience was overall rated

¹¹which depends on each course

as pleasant. It would be interesting not only to rely only on OpenAI models for embedding and textual generation, but also to explore open-source alternatives.

4.3.2 Open Unipa-GPT

In this Section, experiments involving an open-source version of Unipa-GPT are reported: in Section 4.3.2.2 a first comparison was made with different generator LLM based on Llama-2, and a second one with changes in both Embeddings and Generator LLM, along with a fine-tuning phase in Section 4.3.2.3.

Both evaluations were performed with reference to the SOTA baseline as GPT-3.5-turbo as generative LLM, and a suitable test set has been adopted (Section 4.3.2.1), which consists of a subset of questions posed directly by the users during SHARPER Night 2023. Quantitative evaluation was performed, as *golden labels* were manually distilled to build the test set extract from the true user questions. Of course, the University domain did not require a domain expert for this task.

To evaluate the generated answers, the LLM as a judge approach [158] was used, in which an external LLM (e.g. GPT-4-turbo) is queried as an oracle, to provide an evaluation of the provided textual content. In particular, the RAGAS¹² framework [31], a Python library that provides tools to evaluate LLMs applications, was used. In particular diverse AI-based metrics are considered to properly evaluate the generated text, such as context relevance, faithfulness, and correctness of the answers. All scores related to these metrics are obtained by asking the judge LLM model to provide a score to the given input, thus measuring some core characteristics. Due to the generative approach, this inference may result in non-valid scores, which are by default set to zero.

As for *Context relevancy*, the relevancy of the documents retrieved (context) with respect to the question itself is measured. The values fall within the range of (0, 1), where higher values indicate a better relevancy.

Faithfulness measures the factual consistency of the generated answer against the given context, while *answer correctness* involves gauging the accuracy of the generated answer when compared to the ground truth. Both metrics range from 0 to 1 and better performances are associated with higher scores.

4.3.2.1 Evaluation data set

To assess the capabilities of open-source models with respect to OpenAI ones, a small data set was extracted from the 165 collected questions during the SHARPER Night 2023. In particular, for each question typology, as reported in Figure 4.5.a,

¹²<https://docs.ragas.io/en/latest/index.html>

one question was selected, thus obtaining a small set of 6 questions, in which the off-topic category was excluded. This creation process allowed the testing of the developed models over questions from real users and not synthetic ones. Moreover, the small number of selected queries made the manual creation of the relevant associated documents from which a golden answer can be extracted. With the golden answers, it is possible to make a more detailed comparisons with the AI-generated answers. Golden answer were elicited by the document in the Unipa-corpus, and all documents from which the correct answer can be derived are marked as relevant documents. The evaluation data set is shown in Appendix in Table 7.7 (Table 7.8 in Appendix for the English version).

4.3.2.2 Partial Open Unipa-GPT

During the first semester of 2024 more experiments were conducted to evaluate the quality of the answers provided by the Unipa-GPT framework, using different open-source LLM as Generator LLM for the given architecture. GPT-3.5-turbo was considered as a baseline model and open-source models averaging 7B parameters, both foundation- and fine-tuned models with chat- or instruction-oriented strategies, or Italian adaptation were considered.

Llama-2 [133] was selected as a foundational model, along with LLaMAntino-2 [8], its Italian adaptation. For chat-oriented models, Llama-2-chat [133], LLaMAntino-2-chat [8], Fauno [5], were considered, and Camoscio [115], as for the instruction-tuned model. All models are built as fine-tuned versions upon Llama 2, in particular, LLaMAntino-2, Fauno, and Camoscio are obtained from Llama-2, while LLaMAntino-2-chat is a build upon Llama-2-chat.

Evaluation was performed through an instruction prompt using the custom instruction, and the models were forced to generate a maximum of 256 new tokens. Except for GPT-3.5-turbo, all evaluation runs were performed on a 48 GB NVIDIA RTX 6000 Ada Generation without any quantization. Even if the same prompt was used for all the models, it was provided differently for each of them following its authors' specific prompt strategies for either fine-tuning or inference. As an example, *### Istruzione:* was added before the prompt for Camoscio, Llama-2, and LLaMAntino-2, while the [/INST] tag was used for Llama-2 chat. Furthermore, exactly the same retriever with text-embedding-ada-002 was used in all runs; thus, each model is queried with the same documents. The objective is to measure the generation capabilities of the model in terms of phrase and re-elaborate the provided context for generating a helpful answer. All models have been queried with the same generation hyperparameters, which include a low temperature.

The proposed evaluation task is an OQA one where, despite a golden answer provided for a given question, different correct answers can be proposed with different register nuances, according to the Italian diaphasic variation [10]. This means

Table 4.11: BLEU and ROUGE-L scores over baseline and open-source models as Generator LLM.

	BLEU Scores						ROUGE Scores					
	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	<i>Q4</i>	<i>Q5</i>	<i>Q6</i>	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	<i>Q4</i>	<i>Q5</i>	<i>Q6</i>
<i>Baseline models</i>												
GPT-3.5-turbo	0.33	0.15	0.31	0.21	0.21	0.20	0.47	0.30	0.42	0.30	0.19	0.28
<i>Foundation models</i>												
Llama-2	0.0	0.0	0.0	0.0	0.0	0.0	0.05	0.05	0.01	0.01	0.02	0.01
LLaMantino-2	0.0	0.0	0.0	0.0	0.0	0.0	0.04	0.04	0.05	0.01	0.05	0.04
<i>Chat-oriented fine-tuned models</i>												
Llama-2-chat	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.01	0.01	0.01	0.01	0.02
LLaMantino-2-chat	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.05	0.04	0.0	0.0	0.0
Fauno	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.05	0.01	0.01	0.03	0.05
<i>Instruction-tuned models</i>												
Camoscio	0.0	0.0	0.0	0.0	0.02	0.0	0.01	0.02	0.02	0.03	0.03	0.02

that the same concept can be expressed according different linguistic registry as the different person to whom a speaker is talking, leading to the usage of different words and synonyms. In addition, the Italian language has a large linguistic richness as for its lexicon, which leads to a large possibility to express the same concept with different words. Following those linguistic considerations, when a provided sentence needs to be evaluated, an exact match with the reference golden answer cannot be expected, thus obtaining zero as score. In contrast, the fact that it does not happen is not necessarily connected to a completely wrong answer.

On this basis, evaluation of the generated text with reference one can be assumed to be *strict* and *light*. The former involves traditional QA and translation metrics such as BLEU [100] and ROGUE-L score [75], the latter uses novel metrics which involves as external LLM as a judge [158] as in the RAGAS framework [31].

It is worth noticing that BLEU and ROUGE are not specifically tailored for evaluating a text generation task, but they can be considered as a good starting point in automatic evaluation, given a golden answer. They are natively used for automatic translation applications, and their score is linked to n-gramps overlapping between the predicted and reference answers.

The results obtained are reported in Table 4.11 and they show clearly that GPT-3.5-turbo performs significantly better than the foundation- and chat-oriented fine-tuned models that attain 0. Only instruction-tuned models obtain a nonzero BLEU score, but such values remain far low compared to the results for the GPT-3.5-turbo. The results for the ROUGE-L score confirm the comments mentioned above for BLEU: GPT-3.5-turbo performs significantly better compared to other

models, and the foundational models reach the highest results compared to the others. In the context of these traditional metrics, the large quantity of zero values is connected to the fact that an exact match with the golden answer can not always be found, and it can also be associated to the peculiar linguistic characteristics of the Italian language.

A manual inspection of the answers provided by each model was performed to better address the large zero values. Although the low values for GPT-3.5.turbo are caused by linguist differences, the extremely low and zero values for the other models have a different reason. More in detail, a tendency towards generating correct English answers (Llama-2-chat), out-of-topic answers bot in Italian (LLaMantino-2 models) and other languages (Llama-2 and Camoscio) or including code (Fauno) can be found. It is also clear that those generated answers cannot provide different results when compared to the golden answer in a strinct evaluation metric such as BLEU and ROUGE scores.

Further evaluations were carried out using RAGAS to calculate context relevance, faithfulness, and correctness of the responses generated. Since RAGAS metrics need a “judge”, GPT-4-turbo was chosen because it is the most performative model at the time of the experiments.

The results of the Context Relevancy are reported in Table 4.12: scores in each row refer to the first four documents retrieved by the retriever to answer the corresponding question. The starting values mark, for each row, the manually labeled “golden documents”, that were used for generating the corresponding golden answers.

Table 4.12: Context Relevancy scores over baseline and open-source models as Generator LLM. Higher values are marked in bold, underlined values marks the golden documents.

	D1	D2	D3	D4
Q1	<u>0.10</u>	0.10	0.08	0.07
Q2	<u>0.09</u>	0.13	<u>0.63</u>	0.33
Q3	<u>0.10</u>	0.11	0.11	0.11
Q4	<u>0.17</u>	<u>0.06</u>	0.10	0.33
Q5	<u>0.43</u>	0.50	0.14	0.11
Q6	1.0	<u>0.10</u>	0.17	0.50

The results are extremely low, and a correct retrieval (i.e. the document with highest context relevancy coincides with a golden document) is verified in just three of the six considered questions. This finding could be due to the large size

Table 4.13: Faithfulness and Answer Correctness scores over baseline and open-source models as Generator LLM.

	Faithfulness Scores						Answer Correctness Scores					
	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	<i>Q4</i>	<i>Q5</i>	<i>Q6</i>	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	<i>Q4</i>	<i>Q5</i>	<i>Q6</i>
<i>Baseline models</i>												
GPT-3.5-turbo	1.0	1.0	1.0	1.0	0.33	0.88	0.62	0.73	0.56	0.40	0.53	0.67
<i>Foundation models</i>												
Llama-2	0.08	0.0	0.0	0.0	0.0	0.0	0.21	0.18	0.18	0.17	0.18	0.29
LLaMantino-2	0.0	0.0	0.0	0.0	0.0	0.0	0.19	0.19	0.18	0.19	0.19	0.19
<i>Chat-oriented fine-tuned models</i>												
Llama-2-chat	0.0	0.20	0.0	0.0	0.0	0.0	0.18	0.31	0.18	0.20	0.19	0.22
LLaMantino-2-chat	0.0	0.0	0.0	0.0	0.0	0.0	0.19	0.18	0.18	0.19	0.19	0.19
Fauno	0.0	0.0	0.0	0.0	0.25	0.29	0.18	0.21	0.15	0.16	0.18	0.22
<i>Instruction-tuned models</i>												
Camoscio	0.0	0.0	0.0	0.0	0.0	0.0	0.29	0.18	0.17	0.17	0.18	0.18

of the document compared to the length of the question and the actual relevant piece of information. These values, since they are computed on the questions and retrieved documents, are shared between all the different implemented models.

Faithfulness and Answer Correctness scores are reported in Table 4.13. The highest faithfulness results are reached by the GPT-3.5-turbo, which confirms the previous results for both BLEU and ROUGE. In addition, most values are at zero (LLaMantino-2, LLaMantino-2-chat, Camoscio), with the exception of Fauno, which reaches comparable performance with GPT-3.5-turbo in Q5. Similar behavior can be found in answer correctness: among open-source models, none of them reaches performance comparable to GPT-3.5-turbo ones, and none emerges with respect to the others. In addition, quality of the generated answer, proved via manual inspection, is really low and unsatisfactory, thus making none of these models suitable for real-world application, at least without a proper fine-tuning or leveraging explicit instruction-tuned models or providing changes in the generation hyperparameters.

4.3.2.3 Complete Open Unipa-GPT

The complete open Unipa-GPT, as its closed version, is developed as a RAG architecture [70], made up of two main components, as shown in Figure 4.7: the retrieval module and the generator module. In contrast with the open version proposed in Section 4.3.2.2, here the retriever LLM is also open-source.

Architecture

Similarly to Closed Unipa-GPT, the Retriever is made up of a FAISS [29] vector database built using the LangChain library. The vector database is filled with the documents belonging to the Unipa-corpus-clear, divided into chunks 1000 tokens long and with an overlap of 50 tokens. Diverse LLMs were used for Embeddings LLM generation: selected models were the best ones according to the Massive Text Embedding Benchmark (MTEB) [92] for IR¹³. Only models that explicitly state that they were trained and tested with Italian data were selected, namely BGE-M3 (BGE) [19], E5-mistral-7b-instruct (E5-mistral) [141], sentence-bert-base-italian-xxl-uncased¹⁴ (BERT-it) and Multilingual-E5-large-instruct (m-E5) [142] .

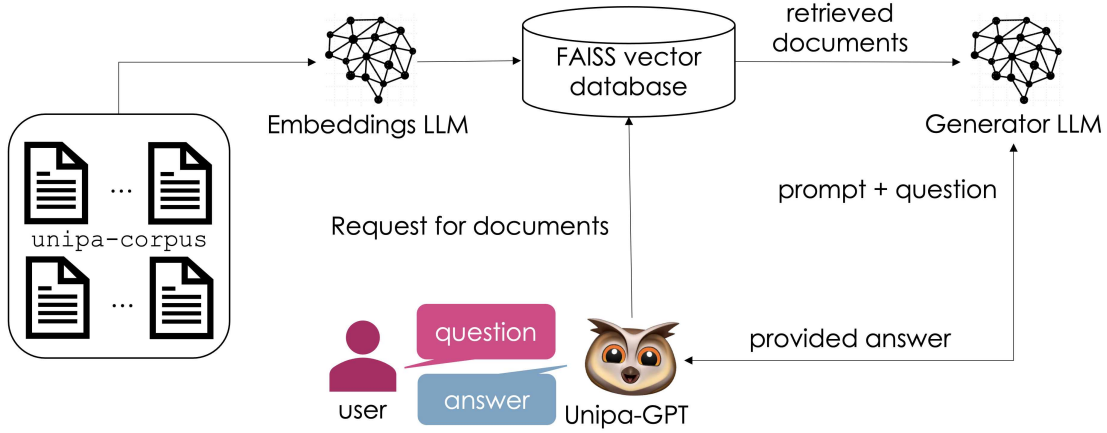


Figure 4.7: Overview of the Open Unipa-GPT architecture

A vector database was built for each model and their corresponding embedding spaces were compared to each other and with text-embedding-ada-002, the OpenAI embedding model, to assess their retrieval performances (Section 4.3.2.3).

As for the generator module, the Italian instruction inserted into the prompt, to let the model to properly answer to user questions:

¹³as in <https://huggingface.co/spaces/mteb/leader-board> in June 2024

¹⁴<https://huggingface.co/nickprock/sentence-bert-base-italian-xxl-uncased>

Sei Unipa-GPT, chatbot e assistente virtuale dell'Università degli Studi di Palermo che risponde cordialmente e in forma colloquiale.
 Ai saluti, rispondi salutando e presentandoti.
 Ricordati che il rettore dell'Università è il professore Massimo Midiri.
 Se la domanda riguarda l'università degli studi di Palermo, rispondi in base alle informazioni e riporta i link ad esse associati;
 Se non sai rispondere alla domanda, rispondi dicendo che sei un'intelligenza artificiale che ha ancora molto da imparare e suggerisci di andare su <https://www.unipa.it/>, non inventare risposte.

The English version is given below.

I am Unipa-GPT, a chatbot and virtual assistant of the University of Palermo, who responds cordially and in a colloquial manner.
 To greetings, answer by greeting and introducing yourself;
 Answer the question with the words "Answer: "
 Remember that the rector of the university is Professor Massimo Midiri.
 If the question concerns the University of Palermo, answer on the basis of the information and provide the links associated with it;
 If you do not know how to answer the question, answer by saying that you are an artificial intelligence that still has a lot to learn and suggest that you go to <https://www.unipa.it/>, do not invent answers.

The proposed prompt is a revised version of the previous one proposed: this resulted in better performances as for a single QA, that is, the configuration used as for those evaluations, instead of a chat simulation.

In these evaluations, newer generative models were considered as Generator LLM, covering both Llama- and Mistral-based architectures. In particular, Llama-3-8B-instruct [82] was used along with its adapted version for Italian, Anita-8B [105], and Minerva-3B [98], which is a mistral-based architecture [53]. All LLMs were evaluated in both their base version and after a fine-tuned version.

Models were fine-tuned with the Unipa-corpus for fine-tuning (Section 4.2.2) with LoRA [49] strategy for 20 and 50 epochs: hyperparameters for LoRA were the same as those used in Alpaca-LoRA¹⁵. In the generation phase, models were asked to output at most 256 tokens. The entire system was developed on a server with 2 Intel(R) Xeon(R) 6248R CPUs, 384 GB RAM, and two 48 GB NVIDIA RTX 6000 Ada Generation GPUs.

¹⁵<https://github.com/tloen/alpaca-lora>

Results

For each QA test pair, the four most relevant documents are retrieved from each vector database built, one for each open Embedding LLM under investigation. Retrieved documents are then judged according to their context relevancy scores with respect to the question provided using the RAGAS framework [31] that takes advantage of GPT-4-turbo for the evaluation.

Table 4.14: Context Relevancy scores over different Embedding LLMs. Higher values are marked in bold, underlined values marks the golden documents.

	Q1				Q2				Q3			
model	D1	D2	D3	D4	D1	D2	D3	D4	D1	D2	D3	D4
open-ai-ada	<u>0.10</u>	0.10	0.08	0.07	<u>0.09</u>	0.13	<u>0.63</u>	0.33	<u>0.10</u>	0.11	0.11	0.11
e5-mistral	<u>0.02</u>	0.03	0.03	0.02	<u>0.50</u>	<u>0.05</u>	0.03	0.03	<u>0.03</u>	0.02	<u>0.02</u>	0.04
bge	0.03	<u>0.02</u>	0.04	0.02	<u>0.05</u>	0.30	0.03	0.09	<u>0.03</u>	0.07	0.04	0.02
bert-it	0.13	0.13	0.03	0.03	0.25	0.33	0.125	0.14	<u>0.18</u>	0.13	0.14	0.02
m-e5	0.13	<u>0.02</u>	0.10	0.08	0.25	0.33	<u>0.50</u>	0.50	0.33	<u>0.02</u>	0.04	<u>0.03</u>
	Q4				Q5				Q6			
model	D1	D2	D3	D4	D1	D2	D3	D4	D1	D2	D3	D4
open-ai-ada	<u>0.17</u>	<u>0.06</u>	0.10	0.33	<u>0.43</u>	0.50	0.14	0.11	1.0	<u>0.10</u>	0.17	0.50
e5-mistral	<u>0.04</u>	0.05	0.05	0.28	<u>0.15</u>	0.04	0.50	0.07	0.33	0.11	0.33	0.11
bge	<u>0.24</u>	<u>0.04</u>	0.33	0.10	<u>0.15</u>	0.04	0.33	0.05	<u>0.18</u>	0.11	0.44	<u>0.33</u>
bert-it	0.15	0.03	0.15	0.03	0.5	0.04	0.04	<u>0.08</u>	0.11	0.33	0.25	0.25
m-e5	0.33	0.50	0.50	0.50	<u>0.15</u>	0.33	0.17	0.5	<u>0.33</u>	0.50	0.11	0.33

Results are reported in Table 4.14, and also include the performances of the original vector database using text-embedding-ada-002 (open-ai-ada) embeddings from OpenAI. The overall scores are not so high, and also the highest relevancy do not always correspond to the golden document used for generating the corresponding answer. In Table 4.14, the underlined values are those associated with golden documents, while the bold values are the highest RAGAS values. A model is considered to perform correctly if the highest context relevance score is assigned to one of the golden documents. This evaluation procedure led to selecting E5-mistral as the best performing Embeddings LLM among the investigated ones.

The superior performances of E5-mistral are also confirmed by a deep analysis on the embeddings space using two different clustering procedures. Embeddings generated by each LLM, starting from documents belonging to both sections *Educational Offer* and *Future Students* of the Unipa-corpus were clustered. The

Table 4.15: SC for each Embedding LLM with the two proposed clustering schemes, degree courses typology and affiliation to a particular Department.

Retriever	SC typology	SC Departments
openAI-ada	-0.10	-0.06
E5-mistral	-0.01	-0.01
BGE	-0.04	-0.07
BERT-it	-0.02	-0.04
m-E5	-0.09	-0.05

embedding spaces were grouped according to either the course degree typology (bachelor/master degree) or the Department where a degree course is affiliated to. Quantitative measures of cluster goodness are reported in Table 4.15, where the Silhouette Coefficient (SC) [113] have been calculated for each model. SC is an internal cluster metric which measures how close a given sample is to other samples in the same cluster and how far away it is from other clusters. Ranges from -1 to 1, and higher values indicate better clustering. Since each document is labeled and native clusters can be defined, in particular one involving the different documents' typologies and section they are referring to, and the other where the different departments are explicitly taken into account in the documents associated within the different course. In the performed evaluation with SC, E5-mistral resulted to be the best performing model, confirming previous insights.

Using the annotated clustering, a scatter plot representation of the embedding spaces for each Embeddings LLM can be drawn (Figure 4.8 and Figure 4.9). The plots have been obtained by 2D dimensionality reduction using t-SNE [136]. They visually expresses and confirm the previous obtained results, where E5-mistral has the best clustering capabilities differentiating within each other both documents according to their typology and the departments they are associated with.

The developed evaluation data set (Section 4.3.2.1) was used to obtain a quantitative evaluation of the correctness of the answers provided by all the Generation LLMs under investigation. Also, answers generated via GPT-3.5-turbo in the original Closed Unipa-GPT set-up were considered as baseline model for comparison reasons.

A complete evaluation of the generated answer is performed in terms of both objective and traditional QA metrics such as the BLEU [100] and ROGUE-L score [75] and novel metrics that utilize the RAGAS framework [31] to evaluate Faithfulness and Answer Correctness. RAGAS metrics are obtained via an external

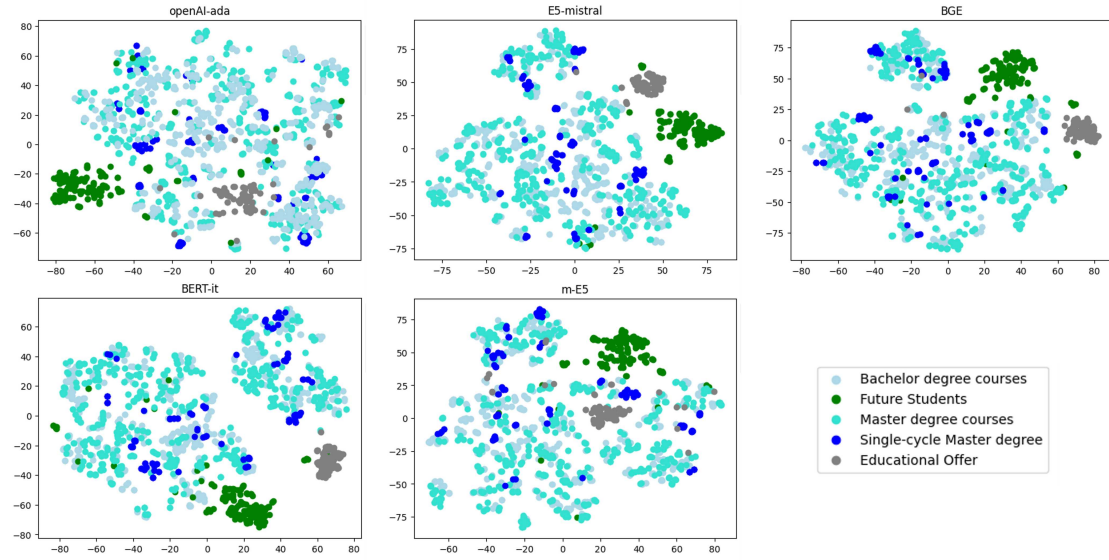


Figure 4.8: Scatter plots of embedding spaces labeled as for typology from the Unipa-corpus

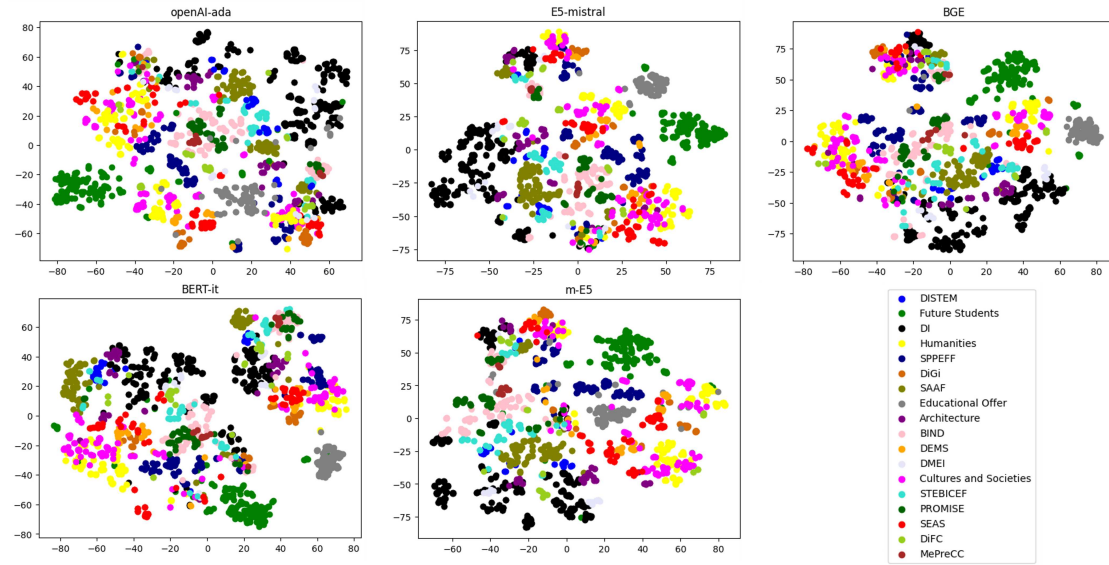


Figure 4.9: Scatter plots of embedding spaces labeled as for department from the Unipa-corpus

LLM acting as a “judge” [158]: in this evaluation, GPT-4-turbo is the considered judge. The results obtained are reported in Table 4.16)

Both BLEU and ROUGE scores are generally low, but this may be mainly

Table 4.16: BLEU, ROUGE-L, Faithfulness and Answer Correctness scores over the generated answers with baseline, open-source, and fine-tuned models

	BLEU Scores						ROUGE Scores					
	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	<i>Q4</i>	<i>Q5</i>	<i>Q6</i>	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	<i>Q4</i>	<i>Q5</i>	<i>Q6</i>
GPT-3.5-turbo	0.37	0.08	0.38	0.20	0.26	0.12	0.51	0.24	0.42	0.30	0.44	0.20
Minerva-3B	0.0	0.0	0.0	0.0	0.0	0.03	0.0	0.10	0.03	0.04	0.04	0.03
Minerva-3B-20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.05	0.01	0.03	0.02	0.04
Minerva-3B-50	0.0	0.0	0.0	0.0	0.0	0.0	0.02	0.01	0.03	0.05	0.018	0.04
Llama-3-instruct	0.05	0.0	0.0	0.0	0.07	0.0	0.13	0.04	0.05	0.11	0.12	0.02
Llama-3-instruct-20	0.0	0.0	0.0	0.0	0.0	0.0	0.01	0.01	0.0	0.02	0.0	0.01
Llama-3-instruct-50	0.0	0.0	0.0	0.0	0.0	0.0	0.02	0.02	0.07	0.04	0.02	0.0
Anita-8B	0.09	0.0	0.05	0.12	0.06	0.05	0.15	0.11	0.10	0.19	0.08	0.09
Anita-8B-20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.08	0.02	0.0	0.0	0.05
Anita-8B-50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.10	0.01	0.04	0.01
	Faithfulness Scores						Answer Correctness Scores					
	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	<i>Q4</i>	<i>Q5</i>	<i>Q6</i>	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	<i>Q4</i>	<i>Q5</i>	<i>Q6</i>
GPT-3.5-turbo	1	0.4	1	1	0.83	1	0.62	0.53	0.74	0.39	0.88	0.40
Minerva-3B	0.0	0.0	0.0	0.0	0.33	0.0	0.21	0.21	0.20	0.21	0.20	0.21
Minerva-3B-20	0.0	0.0	0.0	0.0	0.0	0.0	0.20	0.36	0.19	0.20	0.21	0.60
Minerva-3B-50	0.63	0.14	0.0	0.13	0.0	0.0	0.20	0.21	0.20	0.20	0.21	0.19
Llama-3-instruct	0.5	0.4	0.0	0.0	0.0	0.0	0.44	0.51	0.22	0.44	0.21	0.52
Llama-3-instruct-20	0.0	0.0	0.0	0.0	0.0	0.0	0.18	0.18	0.17	0.20	0.17	0.19
Llama-3-instruct-50	0.0	0.0	0.0	0.0	0.0	0.0	0.18	0.18	0.18	0.18	0.20	0.17
Anita-8B	0.29	1.0	0.83	0.50	0.20	0.18	0.39	0.59	0.35	0.48	0.21	0.41
Anita-8B-20	0.0	0.0	0.67	0.0	0.0	0.0	0.18	0.18	0.19	0.18	0.19	0.20
Anita-8B-50	0.0	0.0	0.67	0.0	0.0	0.0	0.17	0.18	0.19	0.17	0.19	0.19

related to the fact that an exact match cannot be reached between the golden answer and the generated one, and a more semantic comparison should be taken into account. Overall, the answers generated by GPT-3.5-turbo confirm the best ones. By contrast, fine-tuning did not provide the desired improvement in the open-source models: all BLEU scores are almost zero, except Anita-8B. ROUGE scores are higher than the corresponding BLEU ones, and again the base version of each LLM performed better than the fine-tuned ones. Generally speaking, Anita-8B and Llama-3-8B-instruct outperform Minerva, since both reach comparable scores, but the tailored Italian fine-tuning over Llama-3 to obtain Anita-8B it had could be crucial to make it the best performing open-source model during this first automatic evaluation phase.

GPT-3.5-turbo exhibits the best Faithfulness scores despite being surpassed by Anita-8B in question Q2, and also these results confirm the previous considerations about BLEU scores. Something changes in evaluating models in terms of their Correctness: in this case GPT-3.5-turbo is the best model in three answers out of six, followed by Anita-8B (two best results) and Minerva-3B-20 (one best result). It is worth noticing that GPT-based evaluation may lead to a preference over GPT models themselves, but the GPT-4-turbo was the only accessible high-quality generative model at the time of the experiments.

To overcome this issue, the evaluations have been re-executed using gemini-2.0-flash [37] as a judge, and the results obtained are reported in Table 4.17.

Table 4.17: Faithfulness and Answer Correctness scores calculated with gemini-2.0-flash over the generated answers with baseline, open-source, and fine-tuned models

	Faithfulness Scores						Answer Correctness Scores					
	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	<i>Q4</i>	<i>Q5</i>	<i>Q6</i>	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	<i>Q4</i>	<i>Q5</i>	<i>Q6</i>
GPT-3.5-turbo	1.00	0.29	0.93	0.82	1.00	1.00	0.61	0.49	0.89	0.47	0.53	0.33
Minerva-3B	1.00	0.0	0.00	0.0	0.14	0.00	0.19	0.20	0.14	0.19	0.16	0.18
Minerva-3B-20	0.67	0.25	0.00	0.00	0.00	0.20	0.17	0.20	0.16	0.19	0.19	0.33
Minerva-3B-50	0.27	1.00	0.0	0.23	0.00	0.00	0.16	0.20	0.16	0.19	0.19	0.14
Llama-3-instruct	0.11	0.44	0.12	0.25	0.00	0.25	0.44	0.56	0.21	0.37	0.22	0.46
Llama-3-instruct-20	0.00	0.00	0.0	0.00	0.0	0.0	0.15	0.17	0.13	0.16	0.18	0.17
Llama-3-instruct-50	0.00	0.22	0.00	0.00	0.00	0.00	0.13	0.17	0.13	0.0	0.20	0.17
Anita-3	0.11	0.00	0.64	0.50	0.12	0.00	0.36	0.20	0.29	0.42	0.21	0.20
Anita-3-20	0.0	0.00	0.00	0.0	0.0	0.20	0.0	0.18	0.15	0.15	0.18	0.18
Anita-3-50	0.0	0.00	0.00	0.00	0.00	0.0	0.15	0.17	0.15	0.12	0.18	0.18

In this evaluation, the GPT-3.5-turbo outperformed other models, evaluating its better generation capabilities. In this case Minerva and Llama 3 showed interesting performances, but to better evaluate the effectiveness of those evaluations, a manual inspection of the generated answers was performed.

Although it is clear that there are superior capabilities for GPT-3.5-turbo, open-source models answer various tokenization issues, outside topic content, answers in incorrect language and Python content, evaluating the bad results despite the LLM used for the evaluation.

More in detail, the common issue related to the tokenization of the generated output refers to the fact that despite its semantic correctness, the generated text is outputted as a unique word without any spaces, as

Gli elezionidel corsosaracondottoattraversounaprocesso

in Llama-3-8B-instruct, or it is oversplit as

e-domandre d'i-s-c-r-i-z-ion-e-per-l-A.-A.-2023-/-cor-so-n-d'-l-a-u-re-
'-(M-ag-g-is-t-ra-le)-a-dd-ac-ce-o-lib-ro

in Anita-8B. These errors make the models not suitable for human interaction, since it is not possible to read the generated answers, despite the fact that the content is quite pertinent and correct. A deeper analysis on the tokenizer that has been used and a hyperparameter tuning in the generator may lead to an increase of performances.

In addition, models also tend to answer in other languages as

* La durada édié depresso àdue años, * Accesso libre! * Dipartment of
Physics & Chemistry “Emilo Segré” Codice course : 21915

in Llama-3-8B-instruct. This trouble may be related to the intrinsic knowledge of multilingual models, that had been originally trained also over French and Spanish text, is not forgotten despite the Italian fine-tuning. Furthermore, the output languages are linguistically close to Italian and together belong to the Romance Languages [2]: thus, even if the output has to be considered wrong, a linguistic connection can be highlighted.

The most unsatisfactory results are reported for Minerva-3B: the model has a great tendency towards generating answer not related to the given question, and it seems that the answers were generated with samples from the model’s training set. As stated before, a tuning of the generator hyperparameters may help in this case.

Despite the promising results, in some cases the answers by both Anita-8B and Llama-3-8B-instruct are not good from a grammatical point of view, since they are full of mistakes, thus making them not yet ready to be used in real-world applications compared to OpenAI’s ones. Overall results confirm that a (moderate) fine-tuning is not significantly beneficial in terms of performance increase for any model, and, even if it does not reach the same performance, Anita-8B seems to be the most valuable alternative to GPT.

4.4 UniQA

UniQA¹⁶, is an high-quality QA data set that comprehends more than 1k documents and nearly 14k QA pairs. UniQA has been generated in a semi-automated manner using the data retrieved from the website of the University of Palermo, covering information about the bachelor and master degree courses for the academic year 2024/2025. Data are both in Italian and English, which makes the data set suitable for both QA and translation models. To evaluate the proposed data set, a RAG-based model, leveraging on Llama-3.1-instruct as Generator LLM was developed.

In this Section are reported details on the UniQA building process (Subsection 4.4.1), overview of the and experimental setup and results obtained using Llama-3.1 are in Subsection 4.4.2 and 4.4.3, respectively.

4.4.1 Data set

To build UniQA, a web scraping procedure was performed on the website of the University of Palermo¹⁷, using both Selenium¹⁸ and BeautifulSoup¹⁹. Then a collection of a total of 1048 documents, containing information about the bachelor and master degree courses for the academic year 2024/2025, was obtained.

In Table 4.18 the number of documents collected in both the Italian and English splits is reported along with the total (joint). Both Italian and English documents are original, meaning that they have not been obtained via automatic translation tools and have been written natively in both languages. Textual versions of the target documents have been obtained through a scraping procedure from the corresponding pages of the Unipa website either in the Italian or the English version, thus creating the Italian and English split of the data set.

For each available course, two documents have been generated, namely Course info and Course outline, that share an equal header, collecting general information about the course such as the type of degree, the Department of affiliation, and the access rules.

In *Course info* are also reported the educational objectives, the professional opportunities, and the final examination rules for the specific course, while in *Course outline* documents, a complete overview of the teachings, with the corresponding professors, ECTS, and term in which the course is held are listed. Although the

¹⁶<https://github.com/CHILab1/UniQA>

¹⁷<https://offertaformativa.unipa.it/offweb/public/corso/ricercaSemplice.seam>

¹⁸<https://www.selenium.dev/>

¹⁹<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

Table 4.18: Overview of the document splits built from scraping the Unipa website.

	IT-split	EN-split	JOINT-split
Course info	262	262	524
Course outline	262	262	524
Total	524	524	1048

Table 4.19: List of the templates for QA pairs, which leverage the common textual header in each document couple in the English split.

Course info / Course outline - English split	
Questions	Answers
What are the available curriculum for the bachelor degree in course name ?	There are no available curriculum for the bachelor degree in course name .
Is the master degree in course name a open or closed access course?	The master degree in course name is a free access course.
Where do the lessons of the bachelor degree in course name take place?	Lessons are held in location at the department name .
Is it possible to obtain a double degree with a master degree in course name ?	No, it is not possible to obtain a double degree, but it is possible to participate at the Erasmus program.
Provide me some information regarding the bachelor/master degree in course name .	The bachelor degree in course name for the academic year 2024/2025 is a n -year course held in location at the department name . It is possible to choose among one of the following curriculum: curriculum list . It is a closed access course with number seats available. It is possible to obtain a double degree with affiliate university .

University offers a total of 190 bachelor and master degrees, 262 document couples were collected. Having assessed that a course can have multiple *curricula*, which differ from either some classes or the location where the course is held, it was necessary to build both documents for each of them, causing small overlapping and repetitions as for the Course info documents.

Since documents of the same type follow the same architecture, this allows for semi-automated information extraction for the generation of the QA data set. In addition, to each document the following phrases were added: “*For more information visit the course website [link]*”, and “*Per maggiori informazioni consulta il sito del corso [link]*” according to the specific language split and reporting the

Table 4.20: List of the templates for QA pairs, which leverage the common textual header in each document couple in the Italian split

Course info / Course outline - Italian split	
Questions	Answers
Quali sono i curriculum disponibili per il corso di laurea triennale in nome corso ?	Non sono disponibili curriculum per il corso triennale in nome corso .
Il corso di laurea magistrale in nome corso è a numero chiuso o ad accesso libero?	Il corso di laurea magistrale in nome corso è ad accesso libero.
Dove si svolgono le lezioni del corso di laurea triennale in nome corso ?	Le lezioni si svolgono presso la sede di luogo del nome dipartimento .
È possibile conseguire il doppio titolo con il corso di laurea magistrale in nome corso ?	No, non è possibile conseguire il doppio titolo, ma è possibile partecipare al programma Erasmus.
Dammi delle informazioni sul corso di laurea triennale/magistrale in nome corso .	Il corso di laurea triennale in nome corso per l'anno accademico 2024/2025 è un corso della durata di n anni presso la sede di luogo del nome dipartimento . È possibile scegliere uno dei seguenti curriculum: lista dei curriculum . Il corso è a numero chiuso, sono disponibili n posti.

link to the associated web-page of the course.

Table 4.21: List of the templates for QA pairs in the Course info document, English split.

Course info - English split	
Questions	Answers
What are the educational objectives of the master degree in course name ?	Educational objectives from the Course info document.
What are the professional opportunities of the bachelor degree in course name ?	Professional opportunities from the Course info document.
What are the features of the final examination of the master degree in course name ?	Features of the final examination from the Course info document.

Ten different QA templates were generated, five for each language division,

Table 4.22: List of the templates for QA pairs in the *Course info document*, Italian split.

Course info - Italian split	
Questions	Answers
Quali sono gli obiettivi formativi del corso di laurea magistrale in nome corso ?	Obiettivi formativi dal documento course info .
Quali sono gli sbocchi occupazionali che il corso di laurea triennale in nome corso ?	Sbocchi occupazionali dal documento course info .
Quali sono le caratteristiche della prova finale del corso di laurea magistrale in nome corso ?	Caratteristiche della prova finale dal documento course info .

referring to the common header shared by each Course info and Course outline document couple. Templates are reported in Tables 4.19 and 4.20. Moreover, six templates, three for each language split, were generated specifically for each Course info document (Table 4.21 and 4.22), and four templates, two for each language split, for each Course outline document (see Table 4.23 and 4.24).

Table 4.23: List of the templates for QA pairs in the Course outline document, English split.

Course outline - English split	
Questions	Answers
What are the subjects of the target year of the bachelor degree in course name curriculum name ?	Subjects of the target year of the bachelor degree in course name curriculum name are: subjects list . A thesis is also expected to be conducted. It is possible to choose among the following teachings as for optional subjects optional subject list .
Provide me some details regarding the teaching of subject from the master degree in course name .	subject is a n-ECTS subject of the teaching year of the master degree in course name . Teaching is held by professor surname . Lessons will take place during the target semester.

In the last group of templates (Tables 4.23 and 4.24), the former asks for generic information about the list of classes held in a target year, while the latter requests for specific information about a target class. As a consequence, the number of generated QA pairs for each document depends on both the number of years of

Table 4.24: List of the templates for QA pairs in the Course outline document, Italian split.

Course outline - Italian split	
Questions	Answers
Quali sono le materie del anno target del corso di laurea triennale in nome corso nome curriculum ?	Le materie del anno target del corso di laurea triennale in nome corso nome curriculum sono: lista delle materie . È inoltre previsto lo svolgimento della tesi. All'interno del Gruppo di attività formative opzionali è possibile scegliere tra le seguenti materie: lista materie opzionali .
Dammi informazioni sulla materia nome materia del corso di laurea magistrale in nome corso .	nome materia è una materia di n CFU del anno insegnamento del corso di laurea magistrale in nome corso . L'insegnamento è tenuto dal professore cognome . Le lezioni si terranno nel target semestre.

the bachelor or master course and the number of classes themselves in a given year. The following phrases “*For more information visit the course website [link]*”, and “*Per maggiori informazioni vai su [link]*” are further concatenated to each answer of the generated QA pairs in both the English and the Italian split.

It is clear that there are limitations and redundancy on the built data set, with a small amount of manually annotated templates for questions and answers. Despite this, the main focus is towards generating a data set suitable for fine-tuning a LLM for making it able to generate answers, which are not exact frames (spans) of the documents they have been generated from, but a re-paraphrased version, as in a OQA domain. At the end of the generation process, a total of 13742 QA pairs was collected, equally split in 6871 Italian pairs and English pairs, as in Table 4.25.

Table 4.25: Distribution of the UniQA QA pairs in the two language splits.

	IT-split	EN-split	JOINT-split
Course info QA	1.520	1.520	3.040
Course outline QA	5.351	5.351	10.702
Total	6.871	6.871	13.742

4.4.2 Methods

In this Section, the proposed data split for UniQA and the developed experimental setup are reported.

4.4.2.1 Data split

To emphasize the scientific interest of the developed data set, a train-test split of the data set was provided, and their statistics are reported in Table 4.26. Starting from the *Course info* documents, all the unique bachelor and master degrees were first selected, thus courses with multiple curricula were counted once, and a set of 190 courses was obtained. The courses were then subgrouped with respect to the Department they belong to, and for each subgroup a random 80-20 split was done to generate the training and test splits. This procedure was implemented to ensure that:

- bachelor and master courses with multiple *curricula* are considered as a unique block, and then put together in either a train or test split;
- courses belonging to the same Department are equally divided to prevent any bias on the trained models.

In addition, to ensure a coherent split, documents and QA pairs referring to the same degree course are put in the same train-test split across the IT, EN, and JOINT division. To preserve split consistency both for the aforementioned splitting criteria and constraints, and to maintain a significant numerosness in generated splits, a validation set was not defined.

Due to computational constraints on the training procedure of the generation LLM the developed RAG architecture, a *reduced* split of the data set was also created. In the reduced split, only samples whose globally input is less than 3000 tokens are considered. In addition, the created split is 16% smaller than the original one, thus providing a not so significant reduction in performance.

All splits included the QA pairs as well as the original documents, allowing the data set to be suitable for a large variety of NLP tasks, such as translation and QA with support of external knowledge (QA-EK).

4.4.2.2 Experimental setup

A RAG-based architecture was developed to perform QA-EK tasks on the UniQA data set in order to evaluate the quality of the data set itself. In fact, according to the data set generation procedure, both the correctness of the retrieved documents and the generated answers can be calculated, since golden answers are known.

Table 4.26: List of the UniQA train-test splits grouped in Italian, English and Joint split

	Train	Test	All
Documents-IT	398	126	524
Documents-EN	398	126	524
Documents-JOINT	796	252	1.048
QAs-IT	5.298	1.573	6.871
QAs-EN	5.298	1.573	6.871
QAs-JOINT	10.596	3.146	13.742
QAs-IT-reduced	4.293	1.249	5.542
QAs-EN-reduced	5.303	1.556	6.859
QAs-JOINT-reduced	9.596	2.805	12.401

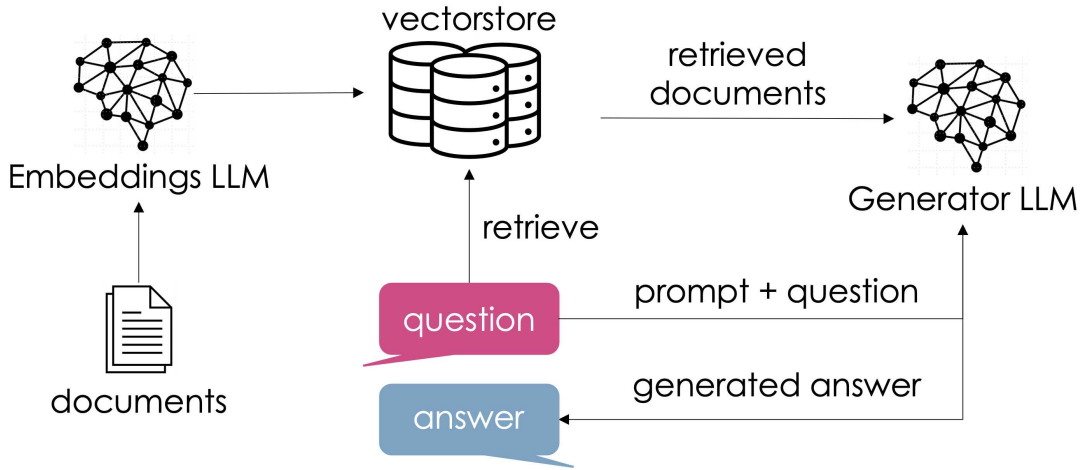


Figure 4.10: Schema of the implemented architecture for QA.

The implemented RAG-based architecture is illustrated in Figure 4.10, which consists of a retriever and a generator module.

The retriever module is composed by a FAISS-based vector store [29], where the generated documents, both from the train and test split, were injected after being split in 1000 token chunks with 100 overlapping tokens, using tiktoken²⁰ as a tokenizer.

²⁰<https://github.com/openai/tiktoken>

The chunks obtained are then processed by an Embeddings LLM, with retriever capabilities that support both English and Italian. Consequently, the best models were selected from the MTEB [92] namely BGE-M3 (BGE) [19], gte-Qwen2-7B-instruct (GTE) [73] and Multilingual-E5-large-instruct (m-E5) [142]. All of them were trained on multilingual data, including English and Italian: actually, it is not so simple to select models that explicitly state that were also trained on Italian data. All are built on the architecture of transformers [137], and both BGE and m-E5 are small 5M models, while GTE is 7B. A vector database for each model was built using the LangChain framework²¹, and their retrieval performances are reported in Section 4.4.3.

As for Generator LLM, Llama-3.1 8B model [82] was selected. The choice of this model was due to its native support for Italian and English as well, and it is an open-source model. Both Foundational and Instruct version were tested and two different English instructions were provided at inference time. *Instruction 1* is designed as the standard instruction prompt and is suitable for Foundational models, while *instruction 2*, follows the instruction prompt suggested for instruction tuning and inference by the authors of the Llama 3.1 models [82]. The instructions are reported in the following.

Instruction 1

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that adequately completes the request.

Instruction: You are Unipa-GPT, the chatbot and virtual assistant of the University of Palermo. Provide a cordially and colloquially answers to the questions provided. If you receive a greeting, answer by greeting and introducing yourself. If you receive a question concerning the University of Palermo, answer relying on the documents given to you with the question. If you do not know how to answer, apologize and suggest that you consult the university website [<https://www.unipa.it/>], do not invent answers. If the question is in English, answer in English. If the question is in Italian, answer in Italian.

Input:

Question: question

Documents: context

Response:

²¹<https://www.langchain.com/langchain>

Instruction 2

< |start_header_id| > system < |end_header_id| >

You are Unipa-GPT, the chatbot and virtual assistant of the University of Palermo. Provide a cordially and colloquially answers to the questions provided. If you receive a greeting, answer by greeting and introducing yourself. If you receive a question concerning the University of Palermo, answer relying on the documents given to you with the question. If you do not know how to answer, apologize and suggest that you consult the university website [<https://www.unipa.it/>], do not invent answers. If the question is in English, answer in English. If the question is in Italian, answer in Italian.

< |eot_id| >< |start_header_id| > user < |end_header_id| >

Question: question

Documents: context

< |eot_id| >< |start_header_id| > assistant < |end_header_id| >

Both instructions were used for querying Foundational models, while instruction 2 was used only for Instruct models. Despite the multilingual task, an English prompt was used, as can be more flexible in a real-world application where the language of the questions and documents provided as input is not known *a priori*.

After evaluating models in their base versions, the best model, which results as Llama-3.1-Instruct, was fine-tuned for 3 epochs. Previous experiments did not show significant improvement for this model in terms of the number of epochs. Fine-tuning was performed with LoRA [49] following the Alpaca-LoRA hyperparameters²² thus UniQA-3-ft was obtained.

The model was trained using the instruction associated (instruction 2), and the golden documents were provided as a context. The entire system was developed on a server with two Intel(R) Xeon(R) Gold 6442Y CPUs, 384 GB RAM, and two 48 GB NVIDIA RTX 6000 Ada Generation.

A further test was performed on the fine-tuned model obtained to assess its generation capabilities in a zero-shot scenario without RAG. In this evaluation configuration, a suitable version of Instruction 2 was devised. Instruction 3 was then defined: it does not contain any mention of relying on external documents for answer generation.

²²<https://github.com/tloen/alpaca-lora>

Instruction 3

< |start_header_id| > system < |end_header_id| >

You are Unipa-GPT, the chatbot and virtual assistant of the University of Palermo. Provide a cordially and colloquially answers to the questions provided. If you receive a greeting, answer by greeting and introducing yourself. If you do not know how to answer, apologize and suggest that you consult the university website [<https://www.unipa.it/>], do not invent answers. If the question is in English, answer in English. If the question is in Italian, answer in Italian.

< |eot_id| >< |start_header_id| > user < |end_header_id| >

Question: question

< |eot_id| >< |start_header_id| > assistant < |end_header_id| >

4.4.3 Results

To evaluate the retrieval performance, a preliminary cluster analysis was performed in the embedding space, based on the “native” cluster labels of the documents, which can be identified in the language of the document (Italian or English) or in the department to which the considered course belongs.

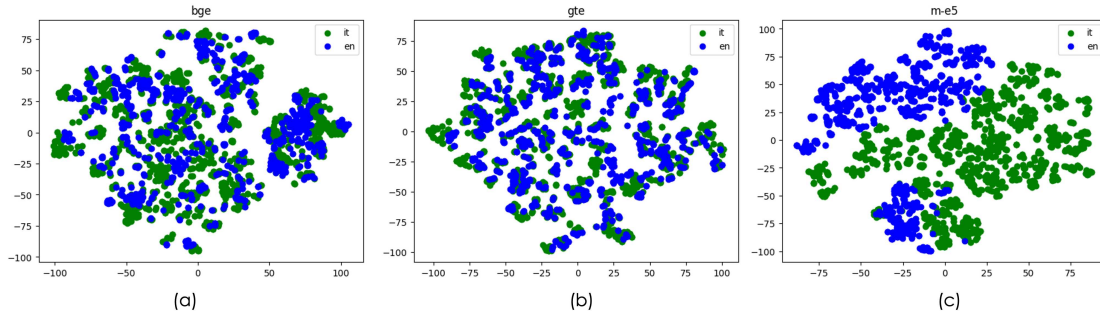


Figure 4.11: Scatter plots of 2D reduced embedding spaces, labeled for each language, from UniQA

The scatter plots of the embedding spaces for each Embeddings LLM are reported in Figures 4.11 and 4.12. Such plots were obtained after a dimensionality reduction performed using t-SNE [136]. In addition to graphical visualization, an analytical analysis, calculating the SC [113] was also conducted, as in Table 4.27.

The graphical and analytical representation highlight that m-E5 has the best clustering capabilities in language separation (Figure 4.11.c) while the other models tend to overlap the embeddings. Conversely, they perform better in grouping

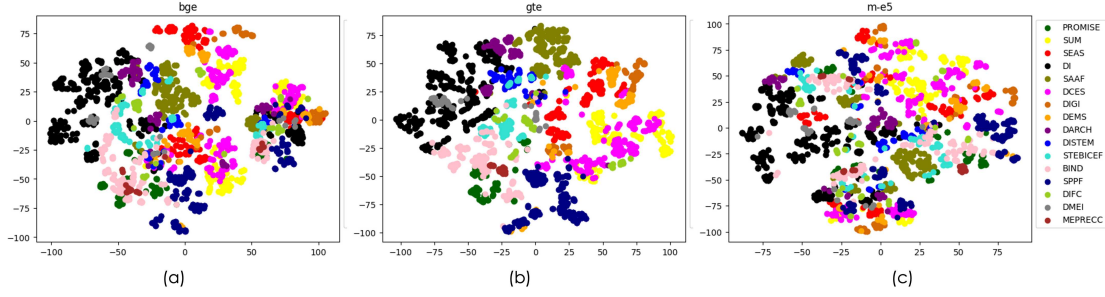


Figure 4.12: Scatter plots of 2D reduced embedding spaces, labeled for each Department, from UniQA

documents in a semantic way, that is, by Department. In particular, GTE outperforms the other models (Figure 4.12.b). Indeed, documents referring to Computer and Mechanical Engineering degree courses, which are taught in the same Department, have much more in common than the ones concerning Nursing or Law. Moreover, the Italian description of a degree course contains many English terms, which can make it harder to cluster documents based on their native language.

The retrieval performances of the models were evaluated by querying their vector stores with a question belonging to the test set: if at least one of the retrieved documents matches the golden one associated with the question, it was considered a correct retrieval. Thus, an accuracy measure was calculated as reported in Table 4.27: here the superiority of GTE is confirmed with an accuracy of almost 86%, while BGE reaches an accuracy of just over 81%, and m-E5 reaches just 77%.

Table 4.27: Clustering and retrieval performances of the three tested embedding LLM for UniQA.

Retriever	SC language	SC departments	Exact matches	Total matches	Accuracy (Exact/Total)
BGE	0.01	-0.09	2.28	2.81	81.39%
GTE	-0.01	0.05	2.41	2.81	85.85%
m-E5	0.22	-0.10	2.16	2.81	76.97%

QA evaluation, was performed by querying the models using the reduced joint version of the test set (Table 4.28) the English only reduced split (Table 4.29), and the Italian only one (Table 4.30). Performances were measured in terms of the BLEU [100] and the ROUGE score [75].

As stated in Section 4.3.2, these metrics are limited since they measure only strict word-by-word similarity and do not take into account language variations,

but represent a well-defined standard in the scientific community to evaluate tasks such as the proposed one. In addition, a manual evaluation of the generated answers is needed and planned as a future work to determine if some changes are needed in the instruction prompt, fine-tuning strategies, or generation parameters. In the context of this work, since the availability of golden references, the attention was towards traditional metrics.

Table 4.28: BLEU and ROUGE score for the different LLMs used for evaluation of the UniQA QAs-JOINT-reduced split.

LLM	Instruction	BLEU	Rouge-1	Rouge-2	Rouge-L	Rouge-Lsum
Llama 3.1	1	0.01	0.09	0.02	0.06	0.06
Llama 3.1	2	0.01	0.07	0.01	0.05	0.06
Llama 3.1 inst	2	0.03	0.16	0.04	0.10	0.13
UniQA-3-ft	2	0.27	0.54	0.38	0.52	0.52
Llama 3.1 retrieved	1	0.01	0.09	0.02	0.06	0.06
Llama 3.1 retrieved	2	0.01	0.03	0.01	0.03	0.03
Llama 3.1 inst retrieved	2	0.01	0.12	0.03	0.08	0.10
UniQA-3-ft retrieved	2	0.16	0.35	0.21	0.33	0.33
UniQA-3-ft no-RAG	3	0.13	<i>0.38</i>	0.20	0.33	<i>0.33</i>

Table 4.29: BLEU and ROUGE score for the different LLMs used for evaluation of the UniQA QAs-EN-reduced split.

LLM	Instruction	BLEU	Rouge-1	Rouge-2	Rouge-L	Rouge-Lsum
Llama 3.1	1	0.01	0.11	0.02	0.06	0.07
Llama 3.1	2	0.01	0.09	0.02	0.05	0.07
Llama 3.1 inst	2	0.03	0.14	0.04	0.09	0.12
UniQA-3-ft	2	0.29	0.58	0.44	0.56	0.56
Llama 3.1 retrieved	1	0.01	0.11	0.02	0.06	0.07
Llama 3.1 retrieved	2	0.01	0.04	0.01	0.03	0.04
Llama 3.1 inst retrieved	2	0.01	0.10	0.03	0.07	0.09
UniQA-3-ft retrieved	2	0.16	0.35	0.22	0.33	0.33
UniQA-3-ft no-RAG	3	0.15	<i>0.42</i>	<i>0.25</i>	<i>0.37</i>	<i>0.38</i>

Inference ability in all models considered was assessed without any quantization strategy. Llama 3.1 Foundational was queried using both Instruction 1 and

Table 4.30: BLEU and ROUGE score for the different LLMs used for evaluation of the UniQA QAs-IT-reduced split.

LLM	Instruction	BLEU	Rouge-1	Rouge-2	Rouge-L	Rouge-Lsum
Llama 3.1	1	0.01	0.07	0.01	0.05	0.05
Llama 3.1	2	0.01	0.05	0.01	0.03	0.04
Llama 3.1 inst	2	0.03	0.19	0.03	0.11	0.14
UniQA-3-ft	2	0.24	0.49	0.30	0.47	0.47
Llama 3.1 retrieved	1	0.01	0.07	0.01	0.05	0.05
Llama 3.1 retrieved	2	0.01	0.02	0.01	0.02	0.02
Llama 3.1 inst retrieved	2	0.01	0.14	0.02	0.09	0.11
UniQA-3-ft retrieved	2	0.17	0.36	0.20	0.34	0.34
UniQA-3-ft no-RAG	3	0.10	0.33	0.14	0.28	0.28

Instruction 2, while Llama 3.1 Instruct and UniQA-3-ft were queried using only Instruction 2, since they are instruction-fine-tuned models.

The evaluation consisted of two runs. In the first run, the golden context was provided to the LLM without the actual retrieve, while the second used the GTE-based retriever module. The former run was aimed at evaluating the model inherent capabilities at generating a correct answer that adheres to the golden one, that is, a re-paraphrase of the context. The latter run aimed to evaluate the end-to-end performance of the whole RAG architecture. Models that make use of RAG, are referred to as retrieved. In addition, tests over UniQA-3-ft in a zero-shot scenario, e.g. without RAG, are referred to as UniQA-3-ft no-RAG, in which Instruction 3 was used.

In Tables 4.28, 4.29 and 4.30, the best results for each run are in bold, while italicized score values have been used in the third run when UniQA-3-ft no-RAG performed better than the UniQA-3-ft retrieved.

The generated answers result in general better in English rather than in Italian, since Llama 3.1 is highly trained in English-based content, despite its multilingualism [82].

As expected, the UniQA-3-ft and UniQA-3-ft retrieved outperform the other models in their respective runs, while their difference in performance is not so significant and depends mainly on the quality of the retrieved documents. Llama 3.1 in its Foundational version, performs a bit better using Instruction 2 with respect to Instruction 1, while Llama 3.1 Instruct clearly shows its ability to follow instructions in both settings. Generally speaking, no fine-tuned models obtained low performances since the generated answer fails to properly adhere to the style of the golden one. In addition, some generation errors are found in the wrong

language used for the provided answer (English content when expecting Italian and *vice versa*) or grammar and spelling mistakes such as

curriculum digital transformation are CORCORATE finance (in english)

in which *CORPORATE* and *ENGLISH* are misspelled, or in

Codicemateria Nome matri Docentecfu Periodossd Attivaformazione
tudentesegreteriatutela

where the model cannot tokenize well. In some cases also out-of-context output is provided, such as textual content in other languages or Python code.

UniQA-3-ft no-RAG reaches performances comparable to those of UniQA-3-ft retrieved, and in some it scores higher than the RAG version. This behavior is mainly addressed to the fact that the fine-tuned model is better able to provide a better adherence to the desired and expected structure with reference to the golden answer. Moreover, the model is not conditioned by the retrieved context that, in the case of UniQA-3-ft retrieved, which may be inaccurate and prone the model towards hallucinations. However, this result is not found in the case of UniQA-3-ft, which, with the correct context, always outperforms both the retrieved and non-RAG versions. In Table 4.31 an example is provided.

This finding clearly indicates that UniQA is a robust high-quality data set that can be used to test both fine-tuned models and RAG architectures. It is important to stress that the structure of the train-test split guarantees that the answers provided by UniQA-3-ft no-RAG leverage only the knowledge acquired during the fine-tuning phase.

In fact, when answering a question belonging to the test set, the model is completely unaware of the degree courses that are not in its training set (Section 4.4.2). In this configuration, the inference capabilities of the model can be truly tested since it is relying on the acquired knowledge from the QA pairs of similar degree courses in the same Department.

Through manual inspection of some of the generated answers, both fine-tuned and no fine-tuned models tend to output misspelled words. Both UniQA-3-ft no-RAG and the non fine-tuned models provide incorrect answers since they have no access to a complete knowledge of the Unipa domain, thus they reply leveraging their native and incomplete knowledge. The non fine-tuned models tend to output verbose answers, and not to provide important information, thus wandering off with a hypothetical course degree outline which is not required and may be imprecise. Generally speaking, the nonfine-tuned models may output some correct information, but in a different format than the one provided in the golden answer,

Table 4.31: Example of a generated answer with fine-tuned models. R- refers to ROUGE Scores.

	Answer	BLEU	R-1	R-2	R-L	R-Lsum
UniQA-3-ft	Subjects for the third year of the master degree in management engineering curriculum digital transformation are CORCORATE finance (in english)[10 points], innovation managem[ement][6 points]. business process manag[ement (in english)]. teaching activities take place during both academic semesters. It is possible t o make electives among the proposed teachings or freecourses.. If it i s necessary further informations please Consult the course web site.http[:] www.uniupa.It/campus/palerno/concorso2017/managment-engineering.htm.	0.10	0.54	0.38	0.52	0.52
UniQA-3-ft retrieved	Subjects of the second year of the master degree in management engineeredigital tranformationarecurricular digital transformation include INDISTRIAL ORGANIZATION AND STRATEGY (in english). teachnigns will continue until June 2018. It is possible touse take optional subjects or subject of your choice.. A final examination takes place during the last semester. It is necessary ti complete all the activities suggested by the syllabus if you want ot obtain the double degree certification issued jointly withevilnis gediminins technical univeristy. For further informations view the officialcoursewebsitehttp://www.unipa.it/struktur.htmid=2026.	0.10	0.35	0.21	0.33	0.33
UniQA-3-ft no-RAG	Subjects for the first semester of the third year of the masterside in managemnt engenering. For more information visit http: //http://www.master.gdaunipd.org/campus/pompeii/ managements.html.	0.00	0.38	0.20	0.33	0.33
Golden answer	Subjects of the second year of the master degree in management engineering curriculum digital transformation are: INDUSTRIAL ORGANIZATION AND STRATEGY (in english), DIGITAL TRANSFORMATION - INTEGRATED COURSE (in english)[SUPPLY CHAIN MANAGEMENT 4.0 (in english), SMART FACTORY (in english), DIGITAL MARKETING (in english)] and LAB OF BUSINESS PROCESS DIGITALIZATION (in english).stage/intership and optional subjects are also included. It is possible to choose among the following teachings as for Optional subjects II:CYBERSECURITY (in english), BUSINESS ANALYTICS (in english), PROGRAMMING, PRINCIPLES OF SMART GRIDS, BIG DATA AND ANALYTICS - INTEGRATED COURSE [MACHINE LEARNING], MACHINE LEARNING [DATA ANALYTICS AND STORAGE (in english)], DATA ANALYTICS AND STORAGE (in english), HR AND CHANGE MANAGEMENT (in english). A final examination is also scheduled.					

thus making it more difficult to evaluate the overall correctness of the generated replies.

In both the golden answers and the retrieved documents, a suggestion for the final user to to visit the website of the degree course to get more information is reported: models try to generate links following the structure of the ones provided in the retrieved documents and in the prompt. The nonfine-tuned models fail since either no link is generated at all or the generated link does not refer to the Unipa website. Fine-tuned models perform better, but not all the generated links are

correct, since misspellings are quite common.

4.5 Discussion

Reported works in this Chapter presents the Unipa-GPT framework, which consists in Unipa-GPT, a virtual assistant capable of answering to secondary school students who need information for accessing and studying at the University of Palermo, and the associated Unipa-corpus. The UniQA data set is also reported, a high-quality QA data set in Italian and English suitable for translation and question-answering tasks where external knowledge is required. An overview of the experiments conducted is reported in Table 4.32.

Table 4.32: Overview of the conducted experiments in Chapter 4.

Experiment description	References
Overview of the Unipa-corpus	Overview of the RAG corpus in Figure 4.1 and Table 4.1. Overview of the fine-tuning corpus in Tables 4.2 and 4.3
Closed Unipa-GPT	Proposed architecture Figures 4.2 and 4.3 and Table 4.4.
Preliminary experiments with Closed Unipa-GPT.	General results overview in Table 4.5. Relevant samples in Tables 4.6, 4.7, 4.8 and 4.9
Experiments with Closed Unipa-GPT during the SHARPER Night.	Developed GUI in Figure 4.4. Prompt used in Table 4.10. General results overview in Table 4.5. Experimental results in Figures 4.5 and 4.6. Small evaluation data set extracted from collected question in Table 7.7
Experiments with partial Open Unipa-GPT in which open-source generator LLM have been tested	Retrieval experimental results in Table 4.12. Generative experimental results in Tables 4.11, 4.13
Experiments with complete Open Unipa-GPT in which both open-source embeddings and generator LLM have been tested	Proposed architecture in Figure 4.7. Retrieval experimental results in Figures 4.8 and 4.9, and in Tables 4.14 and 4.15. Generative experimental results in Tables 4.16 and 4.17.
Experiments over UniQA data set	Overview of the UniQA data set in Tables 4.18, 4.19, 4.20, 4.21, 4.22, 4.23, 4.24, 4.25 and 4.26. Proposed architecture in Figure 4.10. Retrieval experimental results in Figures 4.11 and 4.12 and Table 4.27. Generative experimental results in Tables 4.28, 4.29 and 4.30. Relevant samples in 4.31.

Unipa-GPT was developed leveraging a RAG architecture and an *ad hoc* built documental corpus. Two different versions of Unipa-GPT were implemented: Closed Unipa-GPT with Embeddings and Generator LLMs for OpenAI, and Open Unipa-GPT with open-source LLMs. Evaluations of the developed models were conducted both in an experimental setup and in a real-case scenario, from which a small real data set was expunged for deeper evaluation. Despite the fact that fine-tuning over selected models did not perform as desired, the refinement process over the instruction prompt design was critical for obtaining the best performances. This process, conducted in an iterative manner, included a continuous manual evaluation of the generated answers, during a preliminary experimental phase.

Although the evaluation data set is not so large, it was the only option available to assess, as much as possible, a complete comparison among different LLMs. Moreover, it was possible to manually annotate the golden answer for a more fair comparison between generated answers with traditional and automatic metrics such as BLEU and ROUGE, and the ones based on an external LLM which serves as a judge. While E5-mistral has been demonstrated to be a valuable open-source alternative to OpenAI’s embeddings, none of the considered models attains a generation performance comparable to the ones obtained by GPT-3.5-turbo, even after a fine-tuning procedure. The most promising Generation LLM, which can be used in a real-world architecture, appears to be Anita-8B, but still shows some issues related to both tokenization and grammatical correctness of the output. A deep exploration of different fine-tuning approaches is currently under investigation along with the use of larger open-source LLMs for text generation and generative hyperparameter tuning to increase overall performance.

In the context of this work, significant computational resources are needed to infer such models and fine-tuning them. As GPT models outperformed the open-source one, which are also significantly smaller in terms of parameters, it was necessary to use paid APIs which prevent the usage of local and private instances. This limitation, which is acceptable in the context of this domain in which no personal information is required in the interaction, represents a good trade-off to use closed models and obtain satisfactory answers.

UniQA is a bilingual balanced data set among English and Italian, and was generated without any translation procedure since the documents were natively written in English and Italian, as derived from the Web pages on the Unipa website. UniQA counts 1048 documents and 13742 QA pairs generated in a semi-automated manner.

The potentialities of the data set were tested in RAG-based architecture for QA and capabilities of Llama 3.1 models were also stressed after a fine-tuning

phase. In particular, fine-tuning was crucial to assess superior performances. A further and extensive analysis on UniQA, with the current SOTA for Italian and English LLMs, can be useful to further assess the robustness and performance on UniQA.

In the proposed work in this Chapter, diverse analysis were conducted over the Embeddings and Generator LLM: a study on techniques to improve retrieval via structured output, such as in the form of KG derived from an ontology, can be beneficial for the overall system, as long as an end-to-end fine-tuning of the entire architecture.

Chapter 5

DR-LLM framework

In this chapter are presented the works grouped as DR-LLM [119, 121, 120], involving applications of Visual Language Model (VLM) in medical domain. These works were developed in the context of PNRR FAIR Transversal Project 2: “Vision, Language and Multimodal Challenges”¹, for which the focus was on building multimodal data sets and models for domain-specific applications.

More in detail, the following RQ will be dealt in this Chapter:

RQ 3 *How can we use a multimodal model with domain-based IR for the biomedical domain, and enhanced a decoder-only model to provide a diagnostic suggestion?*

Before building a suitable multimodal model, an appropriate data set needed to be found and re-arranged accordingly, if necessary. MedPix 2.0 data set was built as a reorganization of MedPix[®] [88] (Section 5.2). The obtained multimodal data set contains both visual and textual data related to real clinical cases. Consequently, a classification task was designed to determine the typology of the given medical image and the body part shown. Approaches involving CLIP [106] and BioBERT [69] were explored (Section 5.3). In addition, to enhance model’s capabilities, a domain-based IR was performed, leveraging a RAG-based architecture where decoder-only models such as Minerva [98] and Flamingo [1, 4] were used to build DR-Minerva. On doing this, the prompt given on input to the model was enriched in an in-context learning setup (Section 5.4).

Based on the results obtained previously and to further assess the generation capabilities of models based on decoder-only, such as Llama 3 [82], a hybrid retrieval approach was proposed. More in detail, the retrieval phase in a RAG-based

¹<https://fondazione-fair.it/en/transversal-projects/tp2-vision-language-and-multimodal-challenges/>

setup, was enriched with the use of a novel Knowledge Graph (KG) generated from the textual split of MedPix 2.0, which serves as an external knowledge source [120]. The construction process of KG is based on a RE technique in which a decoder-only model is queried accordingly to extract relevant triplets. For this application, generative models were asked to provide a suggestion of a possible diagnosis using the classification previously obtained, such as scanning modality and body part location, and the retrieved similar cases, arranged in a KG (Section 5.5).

5.1 Related works

The biomedical domain has been explored by the scientific community as a use case for developing and evaluate DL techniques, involving classification and IR. In this context, DL models have to deal with a high-peculiar domain with its own peculiar language used in clinical reports for text-based tasks, and visual characteristics of Medical Imaging for image-based tasks. Despite extensive research in both textual and vision DL models, the medical domain requires an *ad hoc* adaptation that can be achieved through suitable medical data sets.

Medical data sets suffer from various problems, mainly related to the data itself and the peculiarities of the domain. First, there is a privacy issue, since clinical data contain private information about the patient, thus the process of creating such type of data set has to start with an anonymization phase. To overcome this problem, researchers rely on open-access and textbook data, or they collaborate with hospitals to create data sets. The first two methods allow for large scalability, relying on anonymous data. On the other hand, anonymization must be carried out from scratch when dealing with hospitals. In addition, multimodal medical data are scarce compared to other data sets related to different domains such as MS-COCO [76], thus making more challenging training advanced DL models.

One of the most used open-access multimodal data sets for developing medical data sets is PubMed Central[®] (PMC)². This is a widely used free archive of biomedical scientific literature from which it is possible to build one's own data sets via semi-automatic procedures. In PMC, data are anonymized and high-quality captions can be extracted from medical research papers the images belong to. Using the aforementioned process, the following multimodal data sets were extracted from PMC: ROCO [101], MedICaT [129], PMC-OA [77]. ROCO contains pairs of radiology images and the corresponding captions and incorporates an out-

²<https://www.ncbi.nlm.nih.gov/pmc/>

of-class set to improve prediction and classification performances. MedICaT is a disjoint data set from ROCO that is mainly composed of radiology images and provides manually created annotations for subfigures. PMC-OA is larger than the previous ones and maintains a variety of diagnostic procedures, diseases, and findings, while introducing subfigure separation.

VQA-RAD [67] is a data set derived from MedPix[®] [88], and collects a subset of radiological images, while providing QA pairs validated by domain experts.

In addition, textbooks can be a high-quality source of data: PathVQA [43] is a Visual Question Answering (VQA) data set that collects both closed- and open-ended QA pairs, which are extracted from pathology textbooks and digital online libraries through a semi-automated pipeline.

However, open-access data sets such as MIMIC-CXR [55], IU-Xray [25], and SLAKE [78] are manually annotated by domain experts. Both MIMIC-CXR and IU-Xray are chest radiography data sets derived from hospital clinical cases. Both contain a semi-structured radiology report, describing the radiological findings of the images to which it is related. SLAKE, in contrast, collects images from different open-access radiology data sets and provides manual annotations and QA pairs given by experienced doctors in English and Chinese.

INSPECT [50] is a multimodal data set worth mentioning that collects computed tomography pulmonary angiography (CTPA), radiology reports and structured data from EHRS for the diagnosis of pulmonary embolism.

The aforementioned data sets are multimodal ones, and they are deeply focused on VQA tasks. In addition, there are some specialized data sets that provide only visual data, such as UniToChest [18] or free-form textual data [117]. Other works worth mentioning are UniToBrain [35], which integrates a technical report with the scanning modality, and the E3C corpus [85], which is a multilingual data set of clinical reports.

In the first Section of this chapter, the process towards the reorganization of MedPix 2.0, a multimodal biomedical data set derived from MedPix[®] oriented for developing DL models is reported in Section 5.2.

Recent improvements in the development of transformer-based models and the increasing performances of LLMs, able to reach competitive performances with SOTA encoder-only models in most of the traditional NLP tasks, made possible the development of VLM, which leverage textual, visual or audio data. Since Medical Imaging is intrinsically a multimodal domain and focuses primarily on analyzing both images and related reports [42], VLMs are the most used within these applications.

Most of the medical VLM rely on SOTA models, such as CLIP [106], LLaVA [79] and OpenFlamingo [4], an open-source version of Flamingo [1]. Through a fine-tuning procedure of the pre-trained version of these models, researchers succeed in developing their corresponding medical versions, such as BiomedCLIP [150], LLaVa-Med [72], and Med-Flamingo [91]. These models share a common pipeline, in which images and text are encoded separately with their visual and textual models, respectively, and then merged to generate the textual output.

CLIP is trained to learn a multimodal embedding space by jointly training both an image encoder and a text encoder to maximize the cosine similarity of paired image and text embeddings while minimizing the cosine similarity of incorrect pairings [106].

Both LLaVA and Flamingo rely on the pre-trained CLIP visual encoder to extract the visual features from a given image. LLaVA projects the visual features obtained in the word embedding space via a linear layer to pass them to the LM [79]. Analogously in Flamingo, visual tokens are extracted from visual features through a “Perceiver Resampler” and then incorporated with the textual encoding via a cross-attention layer, which is interleaved between the frozen pre-trained LM layers [1].

BiomedCLIP, LLaVa-Med and Med-Flamingo follow the training strategies of the models they are based on, and they are fine-tuned in data sets containing pairs of medical images and their caption, such as MTB [91], PMC-OA and then evaluated with the medical data set for VQA like VQA-RAD, PathVQA and SLAKE. In this context, models are expected to respond to the given question on the basis of the information derived from the proposed image.

In the second Section of this chapter, diverse multimodal approaches were explored, involving CLIP and BioBERT (Section 5.3), a RAG-based architecture with Minerva and Flamingo (Section 5.4) and an innovative hybrid RAG to assess generative capabilities of Llama 3 (Section 5.5).

5.2 MedPix 2.0

Biomedicine research is a fertile field for developing AI-based systems, involving both textual and visual data, in the wide context of Visual Language Model (VLM). To properly train those models, given the critical field of application, they need to be as precise and trustworthy as possible and trained on a suitable high-quality multimodal data set.

Among the suitable data sets in this field, those that contain medical images, such as Computed Tomography (CT) and/or Magnetic Resonance (MR) scans and the corresponding medical reports are not many. One of the most popular data sets is MedPix[®] [88], a free open-access online database of medical images, teaching cases, and clinical topics, integrating images and textual metadata. MedPix[®] includes more than 12,000 patient case scenarios, 9,000 topics, and nearly 59,000 images, but it is not structured and accessible for the development of VLMs.

MedPix 2.0³ is the proposed revised version of MedPix[®], which ensures that there is no privacy-related issue, since it is derived from a freely open-access source and offers a balanced variety of CT and MR scans of different body parts. In addition, for each image, a complete structured clinical case is provided in a textual form, making the data set suitable for multimodal tasks.

In the following Section is reported the process of re-organization of the data set (Subsection 5.2.1), the details on the building process of the curated data set and the proposed task (Subsection 5.2.1.2).

5.2.1 Data set

MedPix[®] [88] is a free open access multimodal online database of medical images, teaching cases, and clinical topics, managed by the National Library of Medicine (NLM) of the National Institutes of Health (NIH). It serves mainly as a support system for Continuing Medical Education (CME) of physicians, nurses, and health-care students. The database collects clinical cases related to more than 12,000 patients. Each case contains at least one medical image and the corresponding findings, discussion notes, diagnosis, differential diagnosis, treatment, and follow-up. Textual information is reported in a semi-structured format. In addition to the clinical case, there is the topic section, where the disease under investigation is discussed in detail from an academic and general perspective.

In Figure 5.1 an example is reported from the MedPix[®] website.

³<https://github.com/CHILab1/MedPix-2.0.git>

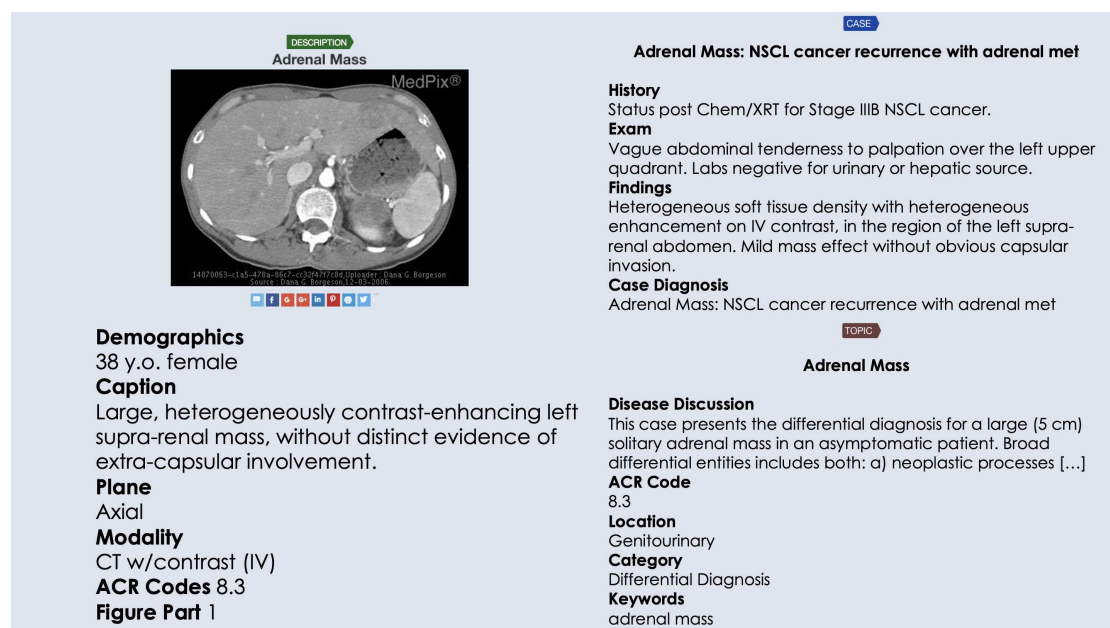


Figure 5.1: Example from the MedPix[®] web site where the original screenshots have been re-arranged for visualizing purposes, and the colored labels have been added for clarity.

Despite the richness of the data set, its free availability, the possibility to add new cases, and the access to clinical cases of interest using keywords, the body part, or the disease, it is not possible to access the raw data. This feature limits the usage of the data set itself for training multimodal AI systems. Therefore, a brand new structured version of this data set is proposed for those applications, because it represents a high quality source for AI-based medical applications. MedPix 2.0 has been built essentially as a MongoDB instance that is released along with a suitable GUI, aimed at general-purpose querying and extracting training data for AI models.

Referring to Figure 5.1, a MongoDB version of the data set was built using a semi-automated pipeline to create two kinds of JSON document: the one collecting the information falling into the screenshots labeled *Description* and the one collecting the information falling into the screenshots labeled both *Case* and *Topic*.

5.2.1.1 Data set extraction

In the context of building MedPix 2.0, only the cases in which were reported CT and MR scans were taken into account from MedPix[®]. The selected images

were downloaded through Open-i⁴. Unfortunately, some noisy images, such as teaching materials or annotated images, are found and, since they are inconsistent as input images for training a VLM, for which a clear CT or MR scan is expected, they were discarded.

A subsequent automatic scraping pipeline was implemented to extract textual data related to the selected images using Selenium⁵ and BeautifulSoup⁶. Finally, two types of JSON documents were devised to store, respectively, the general information related to the clinical case (case-topic document) and those strictly connected with the images (description document).

```
{
  "Type": "CT",
  "U_id": "MPX1009",
  "image": "MPX1009_synpic46283",
  "Description": {
    "Caption": "The prostate is enlarged with several
                calcifications noted within. No dominant
                prostate mass is evident.",
    "Plane": "Coronal",
    "Modality": "CT - noncontrast",
    "ACR Codes": "8.-1",
    "Sex": "male",
    "Age": "73"
  },
  "Location" : "Genitourinary",
  "Location Category" : "Reproductive and Urinary System"
}
```

Figure 5.2: Example of a JSON document in MedPix 2.0 as for *description document*.

More in detail, the *case-topic* document collects clinical cases, identified by their *U_id* code⁷, all patient information, such as age, sex and her/his history, the diagnostic finding and suggested treatment (case information), and general information about the disease from an academic point of view (topic information).

The *description* document collects the caption of the medical images: for each image the corresponding *U_id* code is reported, along with the scanning modality,

⁴<https://openi.nlm.nih.gov/s>

⁵<https://www.selenium.dev/>

⁶<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

⁷unique identifier

```

{
  "U_id": "MPX1009",
  "TAC": ["MPX1009_synpic46283", "MPX1009_synpic46295"],
  "MRI": [],
  "Case": {
    "Title": "Bladder Diverticulum",
    "History": "73-year-old male with hematuria and
      numerous white blood cells found on UA",
    "Exam": "N/A",
    "Findings": "Bladder with thickened wall and
      diverticulum on the right. Diverticulum is
      mostly likely secondary to chronic outflow
      obstruction. Prostate enlargement.",
    "Differential Diagnosis": "Bladder Diverticulum",
    "Case Diagnosis": "Bladder Diverticulum",
    "Diagnosis By": "N/A"
  },
  "Topic": {
    "Title": "Bladder Diverticulum",
    "Disease Discussion": "Bladder diverticula most
      often occur as a result of outlet obstruction.
      Occasionally, a congenital weakness in the
      bladder wall adjacent to the ureteral orifice
      results in a diverticulum. This is termed a
      "Hutch" diverticulum. In children, outlet
      obstruction causing a diverticulum is rare and
      can be seen with urethral [...] .",
    "ACR Code": "8.9",
    "Category": "Diverticulum",
  }
}

```

Figure 5.3: Example of a JSON document in MedPix 2.0 as for *case-topic document*.

the caption of the image and the body part shown. The aforementioned information are reported as key-value pairs following the JSON format: values are textual strings that can be a single word, as for the scanning modality, or a paragraph, as for the differential diagnosis in the case information. An example of the two typology of the created JSON document is shown in Figures 5.2 and 5.3.

In order to properly access the data in MedPix 2.0, a MongoDB database was

Figure 5.4: An example of the interface for querying the database where the user asks for CT scans of female patients (a) and the provided output (b).

created to host all the JSON documents along with the images. The architectural choice stems not only from the nature of the data in MedPix 2.0, but also from considerations related to its high flexibility and scalability. NoSQL database, are suitable for distributed environments, in which private multimodal medical data could be added to the original collections, without being moved away from their generation site, as in the case of hospital generated information.

In addition, a user-friendly interface was built using PyQt5⁸, to allow easy access to the database, querying it to obtain the desired data for visualization and/or download. As shown in Figure 5.4.a, it is possible to select either the collection or the view to be queried and to add the query input in the relative fields. In Figure 5.4.b, an example of the answer to a query is reported: samples matching the query are reported as a list of JSON objects, and the user can save it or view a specific clinical case and/or image selected in the query results list. An example of this view is shown in Figure 5.5, where a MedPix[®] like GUI is reproduced to enforce usability for the users of the original website.

The built interface can be a valid tool for both physicians and AI researchers, since the desired data can be easily downloaded for further application, like training DNN models where a large amount of structured data are required.

⁸<https://www.riverbankcomputing.com/software/pyqt/>

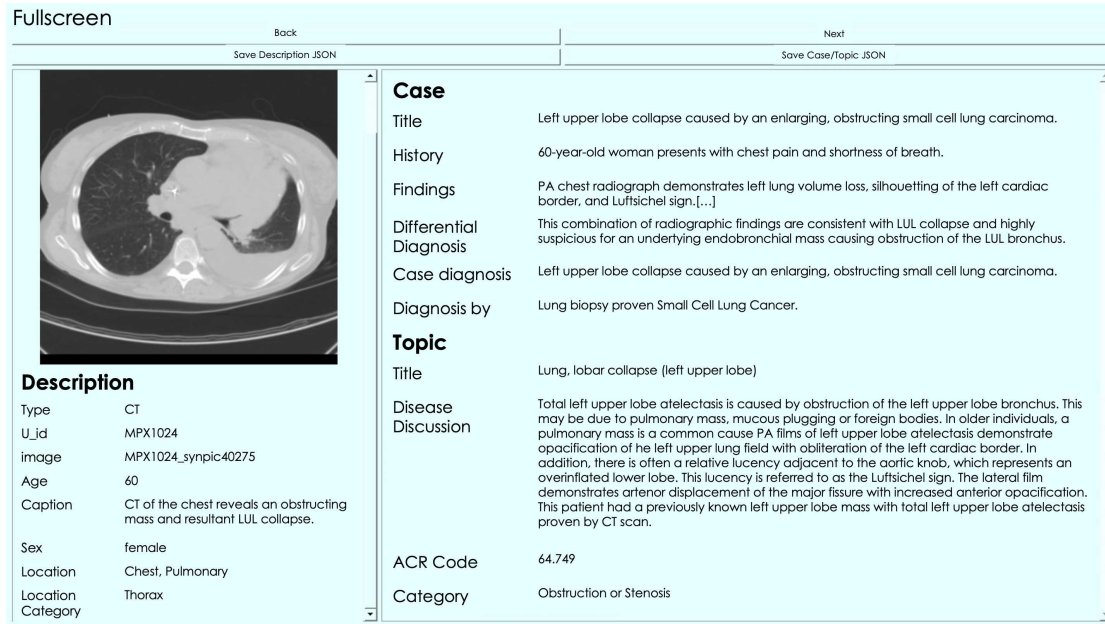


Figure 5.5: An example of the view of a selected clinical case retrieved by the query posed in Figure 5.4. The image, and the relative description are reported along with the information of the clinical case the image belongs to.

5.2.1.2 Data set overview and tasks

Following the illustrated procedure, a total of 2050 medical images, 1062 CT scans and 988 MR scans were derived, along with the corresponding 671 clinical cases (Table 5.1).

Given the data in a more accessible format, it was possible to select the relevant pieces of information and consequently create the proposed classification tasks (or evaluation settings) for the proposed data set. In particular, scanning modality classification, location classification, and joint classification tasks.

Table 5.1: Summary of images and clinical cases for each split in MedPix 2.0.

	CT	MR	Clinical Case
Train	878	775	535
Validation	84	113	67
Test	100	100	69
Total	1062	988	671

Table 5.2: Summary of images divided per macro area of the body for each split in MedPix 2.0.

	Abdomen	Head	RUS	Thorax	SaM
Train	264	742	127	263	257
Validation	23	66	20	30	58
Test	32	76	11	41	40
Total	319	884	158	334	355

The *scanning modality* classification consists of determining whether a given image is a CT or a MR scan, while the *location* requires to individuate the body part shown, and the *joint* setting asks for information about both modality and location. A single location is assigned to each image and refers to the macro-area shown and are Thorax, Head, Abdomen, Reproductive and Urinary System (RUS) and Spine and Muscles (SaM). Despite the more fine-grained field Location is available, (Figure 5.2), in the context of these experiments, the coarse-grained Location Category field is considered, to allow the model a well defined set of options, for which a significant numerosity is available. In MedPix[®] were also reported General and Nervous System labels were also reported, but since samples belonging to these categories were few compared to the others, images belonging to these location were re-assigned by a specialist to the five mentioned. It is also worth mentioning that each image is associated with one location and one scanning modality, while a clinical case can contain multiple images, obtained with different scanning modality and showing different body parts.

To set up the tasks, a semi-automatic Train, Validation, and Test split was built, with an 80:10:10 ratio. Correct stratification of the data required considerable effort, thus ensuring a balance for modality and location labels. In addition, all images referring to the same clinical case are reported in the same split, in order to reduce data leakage. The data obtained from this division are shown in Tables 5.1 and 5.2.

5.3 CLIP over MedPix 2.0

The data set obtained was preliminary tested in the context of modality and location classification, leveraging a CLIP-based model [106] used as a baseline. In particular, CLIP is a multimodal model that succeeds in achieving competitive performance in zero-shot contexts on a wide range of classification data sets by learning the relationships between images and their textual descriptions [135]. CLIP’s architecture consists of an image encoder and a text encoder, and its training scheme involves a contrastive pre-training phase with a large corpus of image-text pairs. After that, the creation of the classifier from the text labels allows inference in a zero-shot setting. An overview of the model is shown in Figure 5.6.

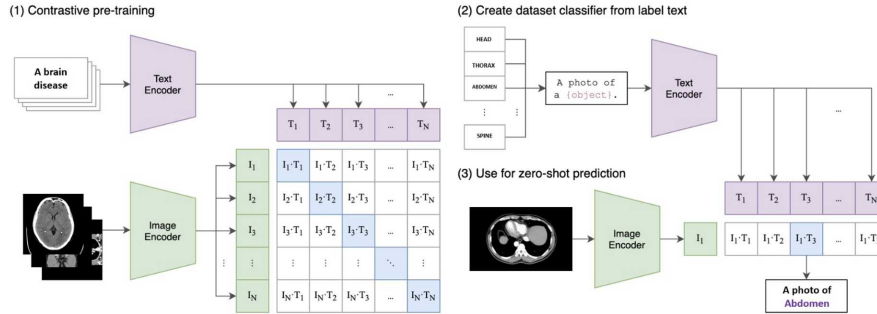


Figure 5.6: Overview of the principal features of CLIP: (1) Contrastive pre-training, (2) Creation of the classifier from text labels and (3) Zero-shot prediction. This schema is a re-adaptation of [106]

The architecture of CLIP is modular and can be adapted to specific visual and textual encoders other than the default ones. Taking advantage of this peculiarity, a trial training phase was started to select the best suited visual encoder for desired purposes, while keeping the textual encoder unchanged. Specifically, a ViT-L / 14 and an RN50x16 were preliminary tested over only the scanning mode task, to select the most promising one for the subsequent phase, which involves fine-tuning of the architecture towards the target domain of the medical imaging with MedPix 2.0. RN50x16 proved to be the best and, combined with the CLIP’s default textual encoder, obtained an accuracy of 58%, which represents an improvement with respect to a random binary classifier which has an accuracy of 50% in this task for discriminate between CT and MR images.

Once the visual encoder had been identified, changes were made to the text encoder and BioBERT [69] was selected to improve the overall classification performance. In addition, to further increase the performance of BioBERT, it has

Table 5.3: Overview of the two inference tasks reporting the used instruction prompt, considered labels, and accuracy score, after fine-tuning

Task	Scanning Modality classification	Location classification
Instruction	This image captured by {label}	This image captured by CT or MR shows Disease located in the {label} area
Labels	CT scan, Magnetic Resonance MR	Head, Thorax, Adbomen, Spine and Muscles, Reproductive and Urinary System
Accuracy	88,00%	52,20%

first been trained on the textual data obtained from MedPix 2.0 via a MLM training strategy, where 15% of the input tokens were masked. The fine-tuned version obtained was integrated with the RN50x16 encoder for further CLIP fine-tuning on the selected train data. Both models were trained for 2500 epochs, with 30 as patience value, using a Cross Entropy loss and Adam [59] as optimizer with a learning rate of $1e-5$.

After this fine-tuning phase, the performance obtained increased considerably, reaching an accuracy of 88% in the scanning modality classification, and 52,50% for the location classification task (Table 5.3). The details of the instruction prompts and labels used for both tasks are reported in Table 5.3.

5.4 DR-Minerva

In this Section a model called Minerva Diagnostic Retriever (DR-Minerva), a Visual Language Model (VLM) specialized in the medical domain and trained in MedPix 2.0 is presented. When prompted to enter a textual input with the patient’s information along with a CT or MR scan, the model provides information about the body part and the scanning modality of the given image. The model relies on the Flamingo [1] architecture, which is well known for its good in-context and few-shot learning capabilities, and it encodes textual data using Minerva LLM [98]. An overview of the developed model is reported in Subsection 5.4.1, along with the experimental phase and related discussion (Subsection 5.4.2).

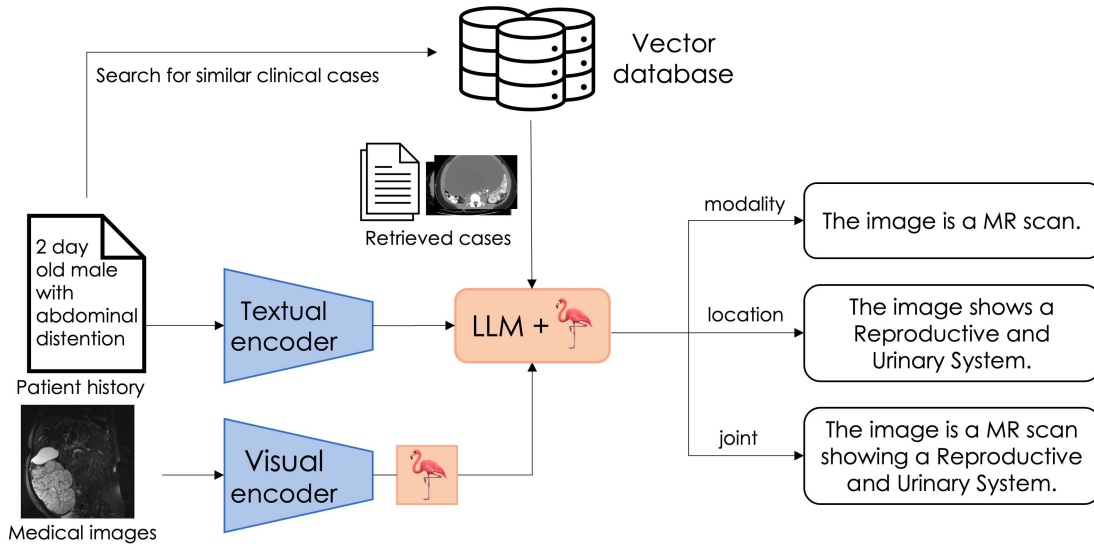


Figure 5.7: Overview of DR-Minerva architecture.

5.4.1 Methods

In the following Subsections, a detailed description of the proposed system is given along with the design choices, and the metrics used for the evaluation phase.

5.4.1.1 Architecture

The overall DR-Minerva architecture is shown in Figure 5.7. The system relies on both the Flamingo architecture [1] and Minerva [98], a new LLM trained from scratch on English and Italian data as part of the activities in the PNRR FAIR Transversal Project 2: “Vision, Language and Multimodal Challenges”. The

project is an effort made by almost twenty Italian Universities. Minerva was chosen because it is a completely open-source model, both training set, architecture, and weights are freely available, and its training set is half in English and half in Italian, making it suitable for further experiments with the developed architecture and Italian data.

Open Flamingo [4], the open-source version of Flamingo, in the 3B parameters version⁹ was used to develop the entire model. Due to the good in-context learning capabilities of Flamingo, it is suitable to be adapted to various domains [1], such as the medical target. CLIP ViT-L/14 [106] was maintained as visual encoder, while Minerva-3B [98] was adopted as language encoder. The bigger version of Minerva, or its Instruction-tuned model, was not taken into account, since experiments were conducted in Summer 2024 and only Minerva base in its 3B version was available. The general model is queried on the MedPix 2.0 multimodal biomedical data set (Section 5.2), using the instruction prompt reported in Table 5.4 to instruct the model how to behave and generate the desired output.

A suitable RAG component was developed for the entire system [70] so, at inference time, the model is provided with an enriched instruction prompt that can improve its performance. Since the AI models are required to be as precise as possible, in particular in medical domain, DR-Minerva was queried with both the target medical image and a template built from some personal information of the patient (e.g. age and sex) followed by the history of the patient in order to prevent empty textual samples. Using the built-in input, the RAG component retrieves (up to) the four closest clinical cases, and they are attached to the prompt as few-shot learning examples.

The RAG module is built via the LangChain framework which in turn uses a FAISS vector database where data are stored using Linq-Embed-Mistral [57]. This is recognized as the best model in MTEB for IR¹⁰.

To effectively evaluate RAG performance, evaluation experiments were run with and without adding the retrieved examples, at inference time. It is worth noticing that in the no-RAG configuration, two examples showing two different scanning modalities are provided to the model to guide it in the generation phase. During the development process, some experiments were performed by querying the model providing just the instruction, i.e. in zero-shot configuration, as reported in Section 5.4.2.

Thus, for each clinical case, the history of the patient, his/her sex and age were considered, and a sample document was created to, along with the annotated

⁹<https://huggingface.co/openflamingo/OpenFlamingo-3B-vitl-mpt1b>

¹⁰as in June 2024

Table 5.4: The structure of the prompt is reported as well as the template of the corresponding expected answers

Type	Instruction prompt	Response
<i>Modality</i>	Given the following medical images and the patient history, provide information about the scanning modality.	The image is a MR scan.
<i>Location</i>	Given the following medical images and the patient history, provide information about the body part shown in the image.	The image shows a head.
<i>Joint</i>	Given the following medical images and the patient history, provide information about the scanning modality and the body part shown in the image.	The image is a MR scan showing a head.

images with modality and location labels, constitute the multimodal context to be retrieved for inference to DR-Minerva. A sample of the data set is shown in Figure 5.8.

Keeping the balance, the training set was further split in train-1 and train-2. Demographics and history of the patient and the clinical diagnosis from train-1, following *ad hoc* designed template, are saved as documents in .txt file, from which the vector database was built. Train-1 serves as retriever corpus at inference time, while samples from train-2 are used for Flamingo fine-tuning. For a pre-training phase on Minerva model, thus reaching an adaptation of the model on the target domain, both training splits were used.

At inference time, the model is queried to provide information on the proposed classification tasks, with the corresponding instruction prompt (Table 5.4). The prompt is further enriched with the retrieved documents and attached images from the train-1 split, and the actual input, made up of an image and a textual document, constructed with a template that reports the sex, age and history of the patient, as in Figure 5.7.

5.4.1.2 Experimental set-up

The whole architecture was developed on server with 96 Intel(R) Xeon(R) Gold 6442Y CPUs and 2 48 GB NVIDIA RTX 6000 Ada Generation.

Minerva was instruction tuned for 20 epochs on a single GPU taking approximately 4 hours, following Alpaca-LoRA setting. Starting from the multimodal split, a training sample was generated by adding to the instruction in Table 5.4, the demographics and history of the patient. Special tokens from Flamingo, namely `<image>` and `<|endofchunk|>` were added and then a training set of 4959 sam-

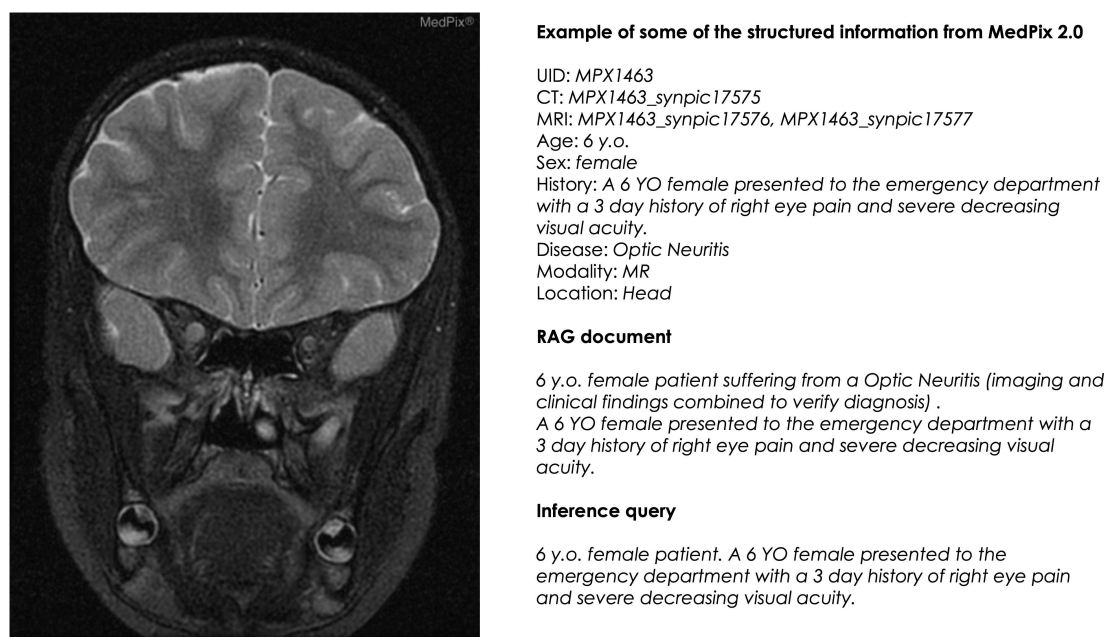


Figure 5.8: Multimodal sample from MedPix 2.0 re-arranged for DR-Minerva with the RAG documents and the inference query.

ples was generated. It is three times larger than the original since samples were created for each of the three tasks, namely modality, location, and joint, which combines both modality and location classification.

Due to computational restrictions, the Flamingo fine-tuning process, with the fine-tuned version of Minerva, was executed over the CPU with a batch size of 1 per 10 epochs, following the train hyperparameters of Open Flamingo¹¹. The whole process last approximately 3 days.

For inference, runs where no RAG context was considered took 20 minutes on average on a single GPU, while those with RAG, approximately one hour and a half, and samples with a larger RAG context that did not fit the GPU, were manually queried on the CPU.

5.4.1.3 Metrics

The created evaluation tasks can be considered as multi-class classification tasks, which performances can be evaluated with classical classification metrics such as accuracy, precision, recall, and F1 score in their macro version are used from

¹¹https://github.com/mlfoundations/open_flamingo

SciKit-learn¹² and reported in Table 5.5.

Table 5.5: Overview of the used metrics in the context of DR-Minerva experiments expressed in terms of TP (True Positive), TN (True Negative), FP (False Positive) and FN (False Negative), and considering C as number of classes.

Accuracy	Precision - macro	Recall - macro	F1 score - macro
$\frac{TP+TN}{TP+TN+FP+FN}$	$\frac{1}{C} \sum_{i=0}^C \frac{TP_i}{TP_i+FP_i}$	$\frac{1}{C} \sum_{i=0}^C \frac{TP_i}{TP_i+FN_i}$	$2 \cdot \frac{P_{\text{macro}} \cdot R_{\text{macro}}}{P_{\text{macro}} + R_{\text{macro}}}$

Since the predicted labels came from a generative model, it is necessary to check if the generated output matches or contains one of the possible labels. If an exact match is found, a valid predicted label can be associated with the generated text; otherwise, an error label is assigned. This assignment is necessary for metrics calculation and it provides the information that the model generates something unmeaning, like a label spelled incorrectly, not generated at all, or invalid. The last is the case in which a plausible label is generated, but does not meet the task constraints. Suppose that inference time for the modality classification task, the model labels an image as a PET (Positron Emission Tomography), which is a possible scanning modality for a diagnostic exam, it would be considered as an error since the task considers only CT and MR as possible scanning modalities. In this sense and in the context of this work, the scanning modality classification task can be considered a binary classification task while the location classification is a multi-class single-label classification. Metrics such as precision, recall, and F1 score are considered in its macro-average. The general idea of the proposed architecture is to improve the results obtained in Section 5.3 which can be considered as the baseline.

5.4.2 Results

In the following Subsections, the experimental results obtained over the test set of MedPix 2.0 are reported, along with a discussion for each task.

As for the evaluation with the fine-tuned version of Flamingo, both the evaluation after 5 epochs (flamingo-ft-5) and after 10 epochs (flamingo-ft-10) are reported. Regardless of the classification task, the model should have at least one single example to generate a meaningful answer. Experiments with only prompt

¹²https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html

and query, not only are unsatisfactory, but also lack of consistency since the model start generating unmeaning output and it is not following the provided instruction. Generally speaking, 5 epochs of fine-tuning Flamingo are enough to reach satisfactory results, and further training can decrease its performance.

5.4.2.1 Modality classification task

Table 5.6: Experimental results for MedPix 2.0 test set in modality classification task. The starred line represent the inference in a zero-shot evaluation mode.

LLM model	Flamingo version	RAG	Accuracy	Precision	Recall	F1 score
Minerva-3B *	flamingo-base *	*	0.17 *	0.17*	0.11*	0.14*
Minerva-3B	flamingo-base		0.36	0.37	0.24	0.28
Minerva-3B	flamingo-base	x	0.31	0.42	0.20	0.19
Minerva-3B ft	flamingo-base		0.41	0.46	0.27	0.34
Minerva-3B ft	flamingo-base	x	0.43	0.32	0.28	0.26
Minerva-3B ft	flamingo-ft-5		0.87	0.89	0.87	0.86
Minerva-3B ft	flamingo-ft-5	x	0.88	0.59	0.59	0.59
Minerva-3B ft	flamingo-ft-10		0.94	0.94	0.94	0.94
Minerva-3B ft	flamingo-ft-10	x	0.82	0.58	0.55	0.55

In Table 5.6 the experimental results are collected over the set of tests for the modality classification task. In this evaluation setting, no significant improvement is provided by the RAG module, while the performances highly benefit from the fine-tuning of Flamingo and the fine-tuning of the model over 10 epochs without RAG, reaches the best performances for every metric considered, reaching at least 0.93 for accuracy, precision, recall, and F1 score. Overall, these performances are not surprising, since this classification task is the easiest one and can be considered as a binary classification task as it is structured. Generally speaking, CT and MR are not the only possibilities for the scanning modality, but in the data set used those are the only two options considered. Following these considerations, the lack of significant improvements in this binary classification with the enhanced context provided by the RAG component, may be addressed for simplicity. Overall, the fine-tuning resulted to be crucial to have significant improvements with respect to the non-fine-tuned and base versions.

Table 5.7: Experimental results for MedPix 2.0 test set in location evaluation setting. The starred line represent the inference in a zero-shot evaluation mode.

LLM model	Flamingo version	RAG	Accuracy	Precision	Recall	F1 score
Minerva-3B*	flamingo-base*	*	0.0*	0.0*	0.0*	0.0*
Minerva-3B	flamingo-base		0.0	0.0	0.0	0.0
Minerva-3B	flamingo-base	x	0.03	0.29	0.02	0.04
Minerva-3B ft	flamingo-base		0.0	0.0	0.0	0.0
Minerva-3B ft	flamingo-base	x	0.04	0.15	0.01	0.03
Minerva-3B ft	flamingo-ft-5		0.29	0.25	0.24	0.14
Minerva-3B ft	flamingo-ft-5	x	0.72	0.52	0.51	0.51
Minerva-3B ft	flamingo-ft-10		0.39	0.37	0.27	0.23
Minerva-3B ft	flamingo-ft-10	x	0.63	0.48	0.46	0.46

5.4.2.2 Location classification task

In Table 5.7 the experimental results are collected over the set of tests for the location classification task. Given the starting accuracy of 0.03 of the base model with RAG, the best results are obtained with Flamingo fine-tuned over 5 epochs, reaching an accuracy of 0.72 while precision, recall and F1 are stable around 0.51. Unlike the modality evaluation setting, this is a harder task for the model. In this case, although performances increase thanks to the RAG usage, highly satisfactory results are not reached, compared to those in Table 5.6, but surprisingly high compared to the starting point. In contrast to the modality classification task, the enhanced prompt significantly improves overall performance, as the task is more complex than the previous analyzed one.

In this case further fine-tuning of Flamingo for 10 epochs is not beneficial for the model; by contrast, textual information provided to the model, along with the corresponding image for location classification, improves and guides the general-purpose visual encoder that does not use any segmentation techniques to analyze the provided images for subsequently classification.

5.4.2.3 Joint classification task

In Table 5.8 the experimental results are collected over the set of tests for the joint classification task, which is not yet considered for the combined prediction task, for the separate one involving only the modality or location. For this reason, accuracy (ACC), precision (PREC), recall (REC), and F1-score (F1) are reported for Modality (M), Location (L), and Joint (J) settings.

Here, it can be clearly noticed how the combined prediction of modality and location and location alone can be difficult for the model, and how it can be substantially improved with the RAG module. It can be seen both as for Accuracy Joint (ACC-J) that grows from 6.50% to 25.50% and as Accuracy Location (ACC-L) from 7.50% to 55.00% without any fine-tuning. As expected, Flamingo fine-tuning leads to the best performances that, as for the joint evaluation setting, reaches an accuracy of only 0.65. This result shows that there is a substantial room for improvement in this classification task and confirms that the multi-label classification task for determining the location of a given images, i.e. the location evaluation, is really challenging for the model despite five macro body parts being considered.

Table 5.8: Experimental results for MedPix 2.0 test set in joint classification task. The starred line represent the inference in a zero-shot evaluation mode.

LLM model	Flamingo version	RAG	ACC-M	ACC-L	ACC-J	PREC-M	PREC-L	PREC-J	REC-M	REC-L	REC-J	F1-M	F1-L	F1-J
Minerva-3B*	flamingo-base*	*	0.25*	0.0*	0.0*	0.19*	0.0*	0.0*	0.17*	0.0*	0.0*	0.18*	0.0*	0.0*
Minerva-3B	flamingo-base		0.20	0.08	0.07	0.13	0.03	0.01	0.13	0.06	0.03	0.13	0.04	0.02
Minerva-3B	flamingo-base	x	0.45	0.55	0.26	0.17	0.52	0.18	0.30	0.40	0.19	0.22	0.43	0.16
Minerva-3B-ft	flamingo-base		0.50	0.27	0.19	0.41	0.20	0.06	0.33	0.18	0.18	0.30	0.13	0.08
Minerva-3B-ft	flamingo-base	x	0.51	0.63	0.30	0.42	0.46	0.28	0.34	0.44	0.20	0.28	0.42	0.17
Minerva-3B-ft	flamingo-ft-5		0.93	0.43	0.37	0.94	0.62	0.46	0.93	0.37	0.30	0.93	0.34	0.24
Minerva-3B-ft	flamingo-ft-5	x	0.89	0.72	0.65	0.60	0.62	0.43	0.59	0.63	0.39	0.60	0.61	0.38
Minerva-3B-ft	flamingo-ft-10		0.93	0.55	0.50	0.94	0.51	0.40	0.93	0.38	0.37	0.93	0.36	0.30
Minerva-3B-ft	flamingo-ft-10	x	0.85	0.68	0.57	0.59	0.51	0.37	0.56	0.50	0.37	0.56	0.49	0.34

5.5 DR-Minerva with Knowledge Graph

In this Section, the previously proposed model DR-Minerva (Section 5.4) is further enhanced with an external Knowledge Graph (KG), built from MedPix 2.0, using Llama 3.1 Instruct 8B [82]. The idea is to improve the generation capabilities of the used models and use the previously obtained results, in terms of modality and location of the given image, to prompt the system in generating a diagnostic suggestions.

An overview of the architecture and data set developed for these tasks is reported in Subsection 5.5.1, while the experimental phase and related discussion is described in Subsection 5.5.2.

5.5.1 Methods

This Section reports on the building process of the architecture based on DR-Minerva as for the joint classification task and the KG used as a retriever to generate a diagnostic suggestion.

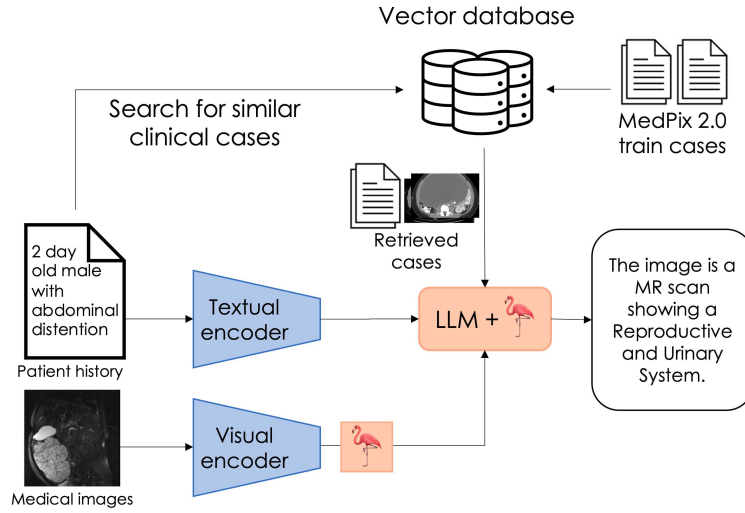


Figure 5.9: Overview of DR-Minerva architecture for the joint classification task.

5.5.1.1 Architecture

Figure 5.9 illustrates DR-Minerva as for the joint classification task. Henceforth, the best model obtained from Section 5.4 is used, that is, the one with CLIP ViT-L/14 [106] as visual encoder, a fine-tuned version of Minerva-3B over the MedPix

Table 5.9: The structure of the prompt is reported as well as the template of the corresponding expected answers

Prompt	Response
Given the following medical images and the patient history, provide information about the scanning modality and the body part shown in the image. <image> <age> <sex> patient. <clinical history of the patient>. [... context ...] The image is a <scan type> scan showing a <body part>. [... context ...] <image> <age> <sex> patient. <clinical history of the patient>.	The image is a <predicted scan type> scan showing a <predicted body part>.

2.0 training set as language encoder, and the Flamingo module fine-tuned per 5 epochs.

Table 5.9 shows the prompt used to query the model in order to obtain the desired response, which consists of both the scanning modality and the body part shown in the input image, along with the sample retrieved that populates the context (as in Section 5.4.1.1).

5.5.1.2 Knowledge Graph definition

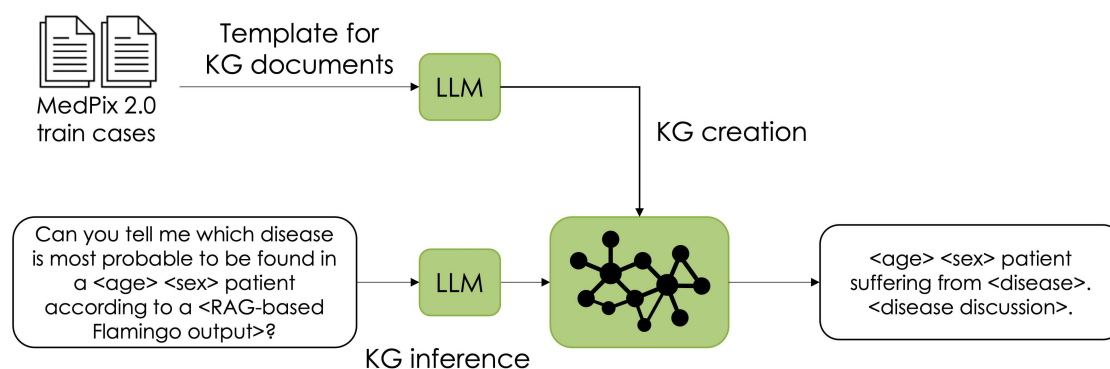


Figure 5.10: Overview of the overall process for KG creation and inference.

MedPix 2.0 can also be seen as a huge archive of historical clinical cases that

can be exploited as external knowledge for sophisticated textual predictions. Here, in addition to information related to the particular clinical case reported in the Case section, such as *Findings* and *Differential Diagnosis*, there are also a broad general information related to the disease, like *Disease Discussion* in the Topic section (Figure 5.3). Due to the inherent domain-specific semantics of this kind of textual information, classical RAG relying on vector databases may not be efficient for precise retrieval in order to generate a diagnosis suggestion, in which cross-references may be helpful. As a consequence, a KG, which relies on both case-specific and general medical information, was built. The created KG is retrieved at inference time to generate a diagnostic suggestion for the input provided. The whole process for creating the KG and inferring suggestions is depicted in Figure 5.10

To allow efficient and effective RE for building the KG, clinical cases belonging to the training set were organized as documents according to the following template:

<U_id> is a clinical report of a <age> <sex> patient suffering from a
 <disease> displayed in <scan modality>.
 <clinical history of the patient>
 The disease <disease name> located in <specific body part> (<body part>).
 <clinical history of the patient>
 <Treatment and followup recommended from the doctor>.
 About <disease> we can say that: <disease discussion>.

Table 5.10: Overview of the hyperparameter setups used to generate the different versions of the KG.

Setup name	max rel. per chunk	Chunks in lower-case
KG-10-N	10	no
KG-10-Y	10	yes
KG-3-N	3	no
KG-3-Y	3	yes
KG-5-N	5	no
KG-5-Y	5	yes

LlamaIndex [80] was used to generate the KG. In particular, the KG is built from document fragments whose fixed size is 4096 tokens. At most n triplets were extracted from each chunk to form the nodes and edges of the KG. Triplets

were extracted using the default prompt proposed by LlamaIndex to Llama 3.1 8B Instruct [82], that realizes a RE system based on decoder-only model and for which neither golden labels nor relation types are known. In addition, Llama 3.1 8B Instruct was chosen for its native multilingual support and its instruction-following capabilities. Six different KGs were created, according to the maximum number of triplets to be extracted from each chunk (3, 5, 10), and the option to process created chunks in lowercase or maintaining in their original form. The hyperparameter setups are reported in Table 5.10, and a simplified overview of the generated KG is reported in Figure 5.11, while a more detailed snapshot is shown in Figure 5.12.

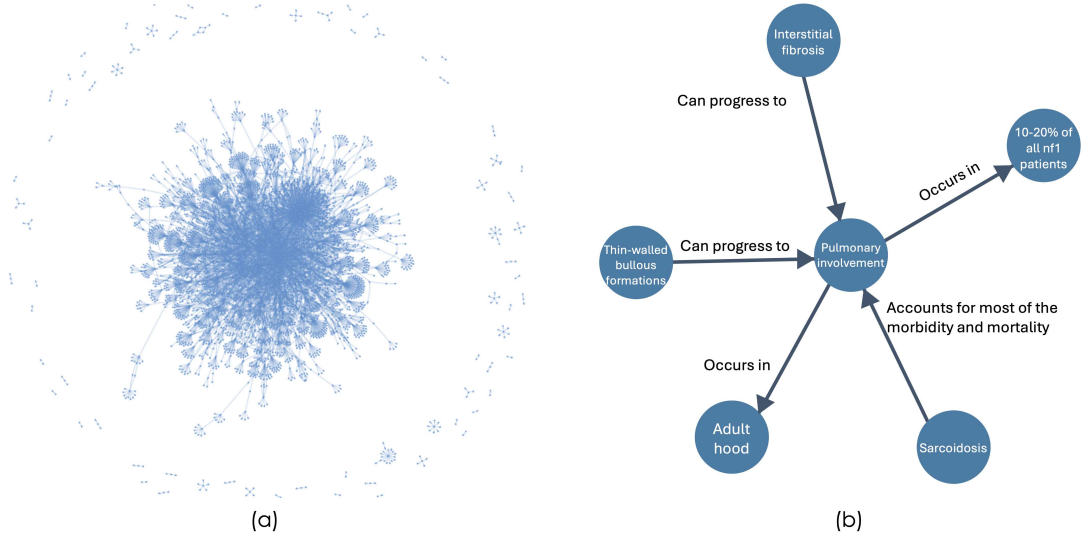


Figure 5.11: Overview of the overall generated KG-10-N (a) and of its subgraph generated from the node “Pulmonary involvement” (b).

It is possible to notice how the rich obtained graph shows some aggregated and clustered areas, while it is not totally connected. This leads to the generative approach according to which it was obtained (Figure 5.11.a). In going deeper into the analysis of the graph, it is possible to notice how the arch contains simple relations extracted from the provided text (Figure 5.11.b) and allows connection across different clinical cases given some intermediate terms as in Figure 5.12, where all diseases that *can be treated with Radiation therapy* are connected to the same node.

The entire process of creation and inference over the KGs was run on a cluster with 2 NVIDIA A100 64 GB GPUs from Leonardo supercomputer¹³ via an ISCRAC application.

¹³<https://leonardo-supercomputer.cineca.eu/it/home-it/>

5.5.2 Results

At inference time, Llama 3.1 8B Instruct is invoked for answer generation by retrieving relevant information from the KGs, leveraging the given question. A low temperature of 0.00001 was used, thus the generative capabilities of the model are limited, and then it is forced to produce an answer as close to the retrieved information as possible. Moreover, to prevent the model from generating loop, the *no-repeat_ngram_size* parameter was set to 2. The query strategy on KG was done using the *tree_summarize* response mode and a value equal to 5 as for the *similarity_top_k* parameter, that is, the default parameters for the retriever.

Performance evaluation was carried out by querying all the built KGs with different hyperparameter choices, using instruction prompts of increasing complexity. Specifically, three instruction prompts were constructed, which were called *simple*, *mid-complex* and *complex* respectively. All the prompts are reported as follows.

Prompt *simple*:

Can you tell me which diseases is most probable to be found in a patient having a $\langle \text{DR-Minerva output} \rangle$?

Prompt *mid-complex*:

Can you tell me which diseases is most probable to be found in a $\langle \text{age} \rangle$ $\langle \text{sex} \rangle$ patient according to a $\langle \text{DR-Minerva output} \rangle$?

Prompt *complex*:

Can you tell me which diseases are the most probable to be found in a $\langle \text{age} \rangle$ $\langle \text{sex} \rangle$ patient according to a $\langle \text{DR-Minerva output} \rangle$? Consider also the following additional information about the patient. $\langle \text{history} \rangle$.

The three instruction prompts use the information provided by DR-Minerva, which consists of the scanning modality and the body part shown in the input image. Instructions were designed to provide more and more information to the KG, as their complexity increases. The maximum complexity will be reached in the last instruction, where the query that a doctor might pose to a diagnostic support system is simulated. In this case, information concerning the patient's clinical $\langle \text{history} \rangle$ is used in addition to the information extrapolated from the images supplied to the multimodal system.

The availability of a test set extrapolated from MedPix 2.0 allowed the generation of a golden answer according to the following template.

$\langle \text{age} \rangle$ $\langle \text{sex} \rangle$ patient suffering from $\langle \text{disease} \rangle$. $\langle \text{disease discussion} \rangle$.

Golden answers were used to evaluate the performance evaluation phase, but despite their availability, defining a suitable metric to evaluate the generated answers, without human judgement, was challenging. Automatic metrics such as BLEU or ROUGE, results inappropriate due to the high variability and specificity in the expected answers.

The generation of a correct clinical text to support a doctor in the diagnostic process may involve that the same meaning can be expressed in a variety of (scientific) forms that depend on many factors: the available knowledge about the particular case when the diagnosis is formulated, the different clinical practices in writing medical records, etc. As a consequence, despite the generated golden answers and the data set splits, it is not reasonable to expect that the model generates text exactly as the golden answer, and in a form that makes scores obtained via BLEU or ROUGE meaningful and coherent.

For these reasons, the proposed evaluation setup is oriented towards embedding similarity scores such as the METEOR [7] and BERT score [151], which was selected because the classical Precision, Recall, and F1 measure can be calculated. METEOR is based on synonym match between the predicted and the golden text, while BERT score compare their embeddings in terms of classical Precision, Recall, and F1 measure, which assess a compliant evaluation with the traditional metrics in AI field. Those scores are intended to assess the degree of overlap of the generated response with respect to the target response, making them well suited to evaluate a specific task of the model, which is to provide a verbose response to the doctor that is as close as possible to the correct clinical terminology.

Table 5.11 shows the results obtained using the three instruction prompts.

The results show that simple prompt achieves the best performance for the models using KG-10-N and KG-10-Y, that are the KGs with the highest number of relations per chunk. The complex prompt performs best for the model using KG-5-Y, and it is close to the simple prompt for the models using KG-3-N. No particular prompt prevails over the others in the models using KG-3-Y and KG-5-N.

Given the structure of the KGs, simple prompt does not constrain the model too much in the generation phase, and it allows the construction of verbose responses. In particular, it is found to be more precise when it is queried for clinical case for which the KG is dense in terms of connection among chunks and relations. On the other end, when the KG is sparse and connects less fragments, the best performance of the prompt complex in terms of BERT-F1 depends on the increased generation precision induced by the insertion of the patient $\langle$ history $\rangle$.

In Table 5.12 a comprehensive overview of the results for the generated KGs is

Table 5.11: Summary of BERT metrics calculated on all KG setups for each of the proposed instruction prompts. Bold values indicate the best score for each KG and body part respectively

<i>KG-setup</i>	<i>Location</i>	Simple			Mid			Complex		
		<i>BERT-Precision</i>	<i>BERT-Recall</i>	<i>BERT-F1</i>	<i>BERT-Precision</i>	<i>BERT-Recall</i>	<i>BERT-F1</i>	<i>BERT-Precision</i>	<i>BERT-Recall</i>	<i>BERT-F1</i>
KG-10-N	Head	0.80	0.78	0.79	0.79	0.78	0.79	0.80	0.79	0.79
	Thorax	0.80	0.78	0.79	0.79	0.78	0.78	0.79	0.78	0.79
	SaM	0.79	0.79	0.79	0.80	0.79	0.79	0.78	0.79	0.79
	Abdomen	0.80	0.79	0.79	0.79	0.78	0.78	0.78	0.78	0.78
	RUS	0.81	0.79	0.80	0.79	0.78	0.78	0.79	0.78	0.78
KG-10-N	Head	0.81	0.79	0.80	0.80	0.78	0.79	0.79	0.78	0.78
	Thorax	0.80	0.79	0.79	0.80	0.77	0.79	0.79	0.78	0.78
	SaM	0.80	0.80	0.80	0.79	0.79	0.79	0.79	0.79	0.79
	Abdomen	0.80	0.78	0.79	0.79	0.78	0.79	0.79	0.79	0.79
	RUS	0.80	0.79	0.79	0.80	0.78	0.79	0.81	0.79	0.80
KG-3-N	Head	0.80	0.79	0.79	0.80	0.78	0.79	0.80	0.79	0.79
	Thorax	0.80	0.78	0.79	0.79	0.78	0.78	0.79	0.79	0.79
	SaM	0.81	0.80	0.80	0.78	0.79	0.79	0.80	0.79	0.80
	Abdomen	0.80	0.79	0.80	0.78	0.77	0.78	0.78	0.78	0.78
	RUS	0.80	0.78	0.79	0.80	0.78	0.79	0.80	0.79	0.79
KG-3-Y	Head	0.80	0.78	0.79	0.79	0.78	0.79	0.80	0.78	0.79
	Thorax	0.80	0.78	0.79	0.79	0.78	0.78	0.79	0.78	0.79
	SaM	0.80	0.79	0.80	0.79	0.79	0.79	0.80	0.79	0.79
	Abdomen	0.80	0.78	0.79	0.80	0.78	0.79	0.78	0.76	0.78
	RUS	0.80	0.78	0.79	0.78	0.78	0.77	0.79	0.78	0.79
KG-5-N	Head	0.80	0.79	0.79	0.80	0.79	0.79	0.80	0.78	0.79
	Thorax	0.80	0.78	0.79	0.79	0.78	0.78	0.79	0.78	0.79
	SaM	0.80	0.79	0.80	0.79	0.79	0.79	0.80	0.79	0.80
	Abdomen	0.80	0.78	0.79	0.78	0.78	0.78	0.80	0.78	0.79
	RUS	0.79	0.79	0.79	0.79	0.79	0.79	0.80	0.79	0.80
KG-5-Y	Head	0.79	0.79	0.79	0.79	0.78	0.79	0.79	0.78	0.79
	Thorax	0.79	0.78	0.78	0.77	0.78	0.77	0.79	0.78	0.78
	SaM	0.78	0.79	0.79	0.80	0.79	0.79	0.81	0.79	0.80
	Abdomen	0.78	0.78	0.78	0.80	0.79	0.79	0.79	0.78	0.78
	RUS	0.80	0.78	0.79	0.78	0.78	0.78	0.80	0.79	0.79

reported, and in addition, a simple baseline is proposed, in which LLM is queried directly without the enhanced context of KG.

In this broad overview, the previous results obtained have also been confirmed, in particular the superior and enhanced capabilities in generating a better answer with retrieval from KG-10 with the simple prompt. As the complexity of the prompt increases, it seems that the additional context and complexity provided to the model at inference time is not beneficial to generate better answers, at least after an automatic evaluation. In addition, the inferred model (Llama 3.1 8B inst) has been queried in its base version (e.g. without any fine-tuning), thus suggesting that focused fine-tuning strategies, coupled with prompt engineering techniques, can improve the overall performances. Undoubtedly, prompt design plays a crucial role in generating the final answer, and improvements in this sense are a future research direction. In this highly specialized and critical domain, the cost of fine-tuning may be carefully considered, since it may provide a significant benefit to the overall model, thus ensuring a better precision.

In the prompt design of the presented works, reasonable rationalities have been considered, along with the acquired experience in the educational domain (Chapter

Table 5.12: Summary of the overall calculated metrics on all KG setups for each of the proposed instruction prompts and a baseline reference. Bold values indicate the best score for each metric

Simple									
	<i>BLEU</i>	<i>ROUGE1</i>	<i>ROUGE2</i>	<i>ROUGEL</i>	<i>ROUGELsum</i>	<i>BERT-Precision</i>	<i>BERT-Recall</i>	<i>BERT-F1</i>	<i>METEOR</i>
Baseline	0.00	0.16	0.02	0.09	0.13	0.76	0.78	0.77	0.12
KG-10-N	0.00	0.20	0.02	0.11	0.17	0.80	0.79	0.79	0.12
KG-10-Y	0.00	0.21	0.02	0.11	0.16	0.80	0.79	0.80	0.12
KG-3-N	0.00	0.20	0.02	0.10	0.16	0.80	0.79	0.79	0.12
KG-3-Y	0.00	0.18	0.01	0.10	0.15	0.80	0.78	0.79	0.10
KG-5-N	0.00	0.15	0.01	0.08	0.13	0.80	0.79	0.79	0.09
KG-5-Y	0.00	0.18	0.02	0.10	0.15	0.79	0.78	0.79	0.11
Mid									
	<i>BLEU</i>	<i>ROUGE-1</i>	<i>ROUGE-2</i>	<i>ROUGE-L</i>	<i>ROUGE-Lsum</i>	<i>BERT-Precision</i>	<i>BERT-Recall</i>	<i>BERT-F1</i>	<i>METEOR</i>
Baseline	0.00	0.15	0.02	0.09	0.13	0.76	0.78	0.77	0.12
KG-10-N	0.00	0.17	0.02	0.09	0.14	0.79	0.78	0.79	0.10
KG-10-Y	0.00	0.17	0.02	0.09	0.14	0.79	0.78	0.79	0.10
KG-3-N	0.00	0.17	0.01	0.09	0.13	0.79	0.78	0.79	0.10
KG-3-Y	0.00	0.17	0.01	0.09	0.14	0.79	0.78	0.79	0.10
KG-5-N	0.00	0.18	0.02	0.09	0.15	0.79	0.79	0.79	0.10
KG-5-Y	0.00	0.19	0.02	0.10	0.15	0.79	0.78	0.79	0.11
Complex									
	<i>BLEU</i>	<i>ROUGE-1</i>	<i>ROUGE-2</i>	<i>ROUGE-L</i>	<i>ROUGE-Lsum</i>	<i>BERT-Precision</i>	<i>BERT-Recall</i>	<i>BERT-F1</i>	<i>METEOR</i>
Baseline	0.01	0.19	0.02	0.11	0.15	0.78	0.79	0.78	0.15
KG-10-N	0.00	0.17	0.01	0.09	0.13	0.79	0.78	0.79	0.10
KG-10-Y	0.00	0.16	0.01	0.08	0.12	0.79	0.78	0.79	0.10
KG-3-N	0.00	0.19	0.02	0.10	0.15	0.80	0.79	0.79	0.11
KG-3-Y	0.00	0.17	0.02	0.08	0.13	0.79	0.78	0.79	0.09
KG-5-N	0.00	0.18	0.02	0.09	0.14	0.80	0.78	0.79	0.11
KG-5-Y	0.00	0.17	0.01	0.09	0.13	0.80	0.78	0.79	0.09

4). As has been reported in the preliminary experiments of Unipa-GPT (Section 4.3.1.2), when dealing with systems that are intended to be used by a specific user, their feedback should be taken into account. In the context of the developed system, physicians should perform a human evaluation of the generated text, both for the correctness and textual quality, and their feedback can be valuable to improve the overall system, thus meeting their needs.

In addition, for each body part, an end-to-end inference is reported using the better prompts and KGs, namely simple prompt and KG-10-N (Figure 5.14) and complex prompt and KG-5-Y (Figure 5.15) are reported, along with the query image, the input prompt, the generated answer, and the BERT-F1 score.

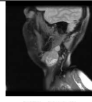
Image	Input prompt w/ age, sex of the patient	Generated answer w/ simple prompt over KG-s1	BERT-F1 score
 MR - Head	51 y.o. female patient.	Based on given information MP 142 and 230 are more related with Head and therefore more probably to find in CT of head. The most probaby disease in this case is MP-142 which is a Glioblastoma multiforme. [...]	0.779
 CT - Abdomen	52 y.o. male patient.	Based on provided context information and prior knowledge about Reporductive and Urinray system diseases, the most likely disease to appear on a MRI scan of the reproductive and urinary system is Urothelial carcinoma. Urotelial carcinoma is a type of cancer that affects the urinary bladder and is the most common type of bladder cancer in adults. It is more common in men than women and typically affects people over the age of 50. [...]	0.803
 CT - Thorax	38 y.o. male patient.	Based on the provided information, we cannot determine the most likely disease to find in patient with CT showing Thoracic region. However, based on general knowledge and the fact that Ewing Sarcoma and Ependyoma are more common in younger patients, it is possible that the patient might have Ewings Sarcomas or Epenymomas. But we need more information to determine which one is more probable.	0.780
 MR - RUS	1 y.o. male patient	The most likely disease to found is renal Cell carcinoma. Reasoning Skill: This question requires the ability to analyze the context and identify the most relevant information. In this case, the question is asking about the probability of finding a disease in an abdomen CT. The answer is based on the fact that the description of mrx1583 mentions that it's a CT w/contrast (IV) of an Abdonem and the location of renal carcinoma is in Genitounary (Reproductive And Urinary System). Therefore, it has a higher probability to have renal Carcinoma. [...]	0.801
 CT - SaM	52 y.o. female patient	Based on context, the disease most probable is Eprendyama. <nlp> Reason: Based on context information from multiple sources, Ependyoma is most commonly found in head and neck region. Therefore it can be concluded that the disease with highest probability of being present at a patient having ct scan showing Head will likely to have ependymoma. [...]	0.793

Figure 5.14: Samples of end-to-end queries to the entire system leveraging KG-10-N and simple prompt.

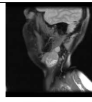
Image	Input prompt w/ age, sex and history of the patient	Generated answer w/ complex prompt over KG-s6	BERT-F1 score
 MR - Head	51 y.o. female patient. HPI: Presented with a 6 month hisotry of an asymptomatic left neck mass s/p failed course of abx therapy and two non-diagnostic FNAs. PMH: Denies any h/o CA, XRT, chemo. Denies any h/o heart dz, lung dz, DM, SLE, scleroderma, [...]	The most probably disease is a benign neoplasm of the neck, specifically a lipomatous tumor. According to all information provided, a lymphadenopathy is less likely, because of two negative FNAB and failed abex therapy. A metastatic disease to neck from a known primary is very unlikely, due to lack of history or symptoms of a primary cancer site, and family cancer history is more suggestive of breast or colon cancer which are less common to metastasize to lymph nodes in the same side of primary. Finally, an inflammatory process is highly unlikely due the lack symptoms and negative history for infections. [...]	0.786
 CT - Abdomen	52 y.o. male patient. 52 yo male with RUQ pain and abnormal liver associated enzymes.	The most probabale disease that could be found in this male is Prostat Cancer. Reason: 1- The age of patinet (52) makes him more likely for Prostrat Cancer 2- He presents symptoms such as difficulty starting or stopping urine flow which are common symptom related To prostata Cancer 3- He has RU pain which can be associated with liver and liver is affected by metastasized Prosta Cancer cells. [...]	0.800
 CT - Thorax	38 y.o. male patient. Thirty-eight year old male presents to primary care complaining of chronic cough and shortness of breath that has significantly progressed over several days	Based on the provided information, it is not possible to determine the most likely disease for a patient with symptoms of cough, short breath, chronic, progressive over days, 37 y. old, male, thorax CT. The information provided is too general and does not provide specific details about any disease that would be most relevant to this patient. However, based on general medical knowledge and the symptoms described, some possible diseases that could be considered include: Pneumonia, Pulmonary Embolism, Lung Cancer, Tuberculosis, etc. [...]	0.807
 MR - RUS	1 y.o. male patient 2 day old male with abdominal distention	Based on the provided information, it is not possible to determine the most likely disease to find in the 4 y. o. male patient. The information provided is about a different patient (0y.o) and a CT- non contrast scan of head. There is no information available about an abdominal CT-scan of this patient or any other patient with the same age and gender. However, based on general knowledge and the additional provided context, the diseases that are most common in children with an age range of around 6 months to 10 years are: Intussusception, Appendicitis, and Gastrointestinal obstruction. [...]	0.803
 CT - SaM	52 y.o. female. 52 yo woman presented with headaches and mental status changes. Hx of L breast SCCa three years ago treated at that time with lumpectomy & chemoradiotherapy.	The most possible disease in this patient, based in the CT Scan and the clinical history, is a metastatic disease from breast Cancer. This conclusion is based in that the woman has a history of breast cancer and has symptoms that are consistent with metastases. Therefore, the final diagnosis is Metatasis.	0.820

Figure 5.15: Samples of end-to-end queries to the entire system leveraging KG-5-Y and complex prompt.

5.6 Discussion

The proposed MedPix 2.0 is a valuable multimodal data set of clinical reports, CT and MR scans, obtained via a semi-automated pipeline through which images were downloaded, selected and textual information was arranged as in JSON files. The data set was derived from MedPix[®] and re-arranged in a suitable format for training AI models.

Although this data set may be close to VQA-RAD (Section 5.1), they differ from both the sampling strategy and the samples themselves. In MedPix 2.0, QA pairs are not integrated, but different QA pairs can be created based on the structured textual information provided in the data set itself. Thus, the creation of more complex tasks, such as document summarization or understanding, is possible. In MedPix 2.0, in fact, the textual data contains both information related to each image, but also general information about the clinical case and the disease itself.

To assess the scientific validity of MedPix, different classification tasks were proposed, namely classification of the scanning modality and body part (location) shown in a given scan, yet as a separated yet as a joint task.

Diverse models were built and tested with the proposed tasks. A first development was with a CLIP-based model, with a custom version of BioBERT, trained with a MLM strategy. In this context, only scanning modality and location classification task were performed: while promising results were obtained for scanning modality, room for improvement is shown in location classification one.

To improve the CLIP-based architecture, DR-Minerva was proposed, a VLM built on Flamingo and a fine-tuned version of Minerva LLM as textual encoder. The developed model benefit of fine-tuning and RAG-based approach, which allowed for a well-structured prompt where cases similar to the input one, were given as a few-shot example in an in-context learning setting. This strategy was a turning point, since the developed model outperformed the CLIP-based one in both modality and location classification tasks. In addition, the joint classification was also tested, which is promising, but it still remains a challenging task.

Using the result obtained, a step further was taken starting from the MedPix 2.0 data set, from which a KG was built starting from a suitable reorganization of the textual documents in the corpus. The KG was created automatically based on Llama 3.1 8B Instruct starting from the clinical cases in the train split, which contains both diagnosis and a general overview of the disease diagnosed. On this basis, a further evaluation was performed: the output from DR-Minerva, combined with the clinical history of the patient, served as a query to retrieve information

from the KG. Obtained information, along with the original input, is passed to Llama 3.1 8B Instruct, to provide a diagnosis suggestion. Numerous experiments conducted show that a well-structured knowledge base can be crucial as a retriever component to enhance the quality and correctness of the final output.

In conclusion, for such critical domain and application, which requires personal and sensitive information, the usage of API-based models cannot be taken into account. In this sense, usage of relatively small models has been explored and their performance can be improved with a suitable fine-tuning procedure. In addition, the pure generative phase with Llama 3.1 8B Instruct coupled with the KG, was performed with the base version of the model, for example, without any fine-tuning. The results obtained suggest that improvements can be found following such an approach or improving both the retrieval component and the KG construction.

An overview of the experiments conducted is reported in Table 5.13.

Table 5.13: Overview of the conducted experiments in Chapter 5.

Experiment description	References
MedPix 2.0 data set details	Original data set in Figure 5.1. Samples from the data set in Figures 5.2 and 5.3. Overview of the developed interface Figures 5.4 and 5.5. Data set statistics in Tables 5.1 and 5.2
Preliminary experiments with MedPix 2.0 with a CLIP-based architecture	Architectural details in Figure 5.6. Experimental results in Table 5.3.
Experiments with MedPix 2.0 with DR-Minerva LLM	Architectural details in Figure 5.7 Re-arranges sample from MedPix 2.0 in Figure 5.8. Prompt definition in Table 5.4. Evaluation metrics in Table 5.5. Experimental results in Tables 5.6, 5.7 and 5.8.
Experiments with MedPix 2.0 with DR-Minerva enhanced with KG.	Architectural details in Figure 5.9 and 5.13. Prompt definition in Table 5.9 KG definition image 5.10 and Table 5.10. Experimental results in Tables 5.11 and 5.12 and Figures 5.14 and 5.15

Chapter 6

Conclusions

In this work, a general overview of the recent improvements in NLP field in context of IR and QA were proposed with focus on transformer-based models. Both analyses on encoder-only models for RE and text classification were proposed, and decoder-only models for complex architectures involving textual and multimodal QA and classification, in RAG setup and in-context learning strategy, were proposed.

For encoder-only-based models, which were the focus of RQ1, no relevant improvements were reported with the proposed approaches, compared with SOTA methods. Proposed methods shares the same macro architecture, involving a LM for embedding extraction, and DNN, such as LSTM, BiLSTM, transformers and dense layers, for further processing, and a classifier.

With focusing on RE task, the developed neural model for SemEval 2010 Task 8, represents the first approach with LMs as BERT and with the idea of developing models that can work in a resource-limited environment. Two different data encodings were proposed and tested among the selected LMs. Unfortunately, none of them outperformed the other, but a fine-tuning result is crucial to obtain satisfactory results, across all LMs and developed networks. Moreover, the so-called concatenated encoding is the most effective on the task due to the usage of the [SEP] token.

Regarding the text classification proposed for EVALITA 2023, the results obtained did not reach significant results in the competitions, but the proposed analysis for both HODI and HaSpeeDe data sets, represent a reflection on the intrinsic nature and ambiguity of the data. Both data sets, in context of social media posts, result challenging to be correct disambiguated, since comments representing a negative opinion are not necessarily hateful, and the usage of slurs can have a (re)appropriation connotation. The results obtained suggest that a semantic and

domain-specific analysis may not be the optimal solution due to the lack of a sharp semantic distinction between the two classes, but may be considered coupled with other methods.

Regarding decoder-only models, two main works were illustrated, namely Unipa-GPT and DR-LLM frameworks, which involve English and Italian LLMs for two specialized domains, namely education and biomedicine, proposing two RAG based approaches.

The main contributions related to the works belonging to the Unipa-GPT framework (RQ2) are two data sets, one is the Unipa-corpus, which involves an Italian document corpus with an automatic generated set of QA pairs, the other is UniQA, an English and Italian data set of documents and semi-automatic generated QA pairs and for which no automatic translation techniques were involved. While Unipa-corpus serves to implement a chatbot-based application, UniQA is focused toward QA applications in which golden answers are available and a more comprehensive and quantitative evaluation can be performed. Despite their differences, both corpora serve to implement a generative application based on a LLM enriched with a RAG-based architecture. Injecting external knowledge at inference time instead of a costly fine-tuning is the optimal choice for a continuing and changing application. Also, the prompt design required diverse testing, to define a proper instruction to be provided to the model, assuring the best answers, as for tone and what to reply on case of out-of-context questions.

Despite the chatbot application being mainly tested and used in the real world during SHARPER Night leveraging GPT models from OpenAI, open-source generative models were also tested and evaluated in an experimental setup. None of the recent released Italian LLMs, tested without any kind of fine-tuning, reached performances comparable to those obtained by GPT models from OpenAI. More in detail, a subsequent evaluation was conducted on a small set of the most recent selected LLMs, in which a fine-tuning procedure confirmed that Anita, an Italian adaptation over LLama 3 in its instruct version, is the most promising generative LLM for the given application. A subsequent definition of a suitable and effective fine-tuning strategy is necessary to reach significant results. Evaluations of retriever components were also conducted, and a valuable alternative to Ada, which is an OpenAI embedding model, is found in E5-mistral.

The works related to UniQA evaluation mainly involved an analysis of Llama 3.1 in various versions. The complete and validated data set, for its multilingual and availability of golden answers, allowed a deeper analysis of the embedding space defined by the documental corpus, establishing that GTE is the most performative one for retrieving purposes. As for the generative component, the per-

formances with Llama 3.1 in its base and instruct versions, and after a fine-tuning procedure, determine the need of a fine-tuning to improve the overall performances for the target application.

DR-LLM represents the multimodal contribution of this work, with a focus on the biomedical domain (RQ3). In this context, the re-arrangement of MedPix[®], allowed the creation of a multimodal data set in the medical domain, MedPix 2.0. Leveraging the obtained data set, further experimentation was conducted involving both textual and visual encoders with a focus on the main challenging tasks: classification of the typology of the given medical image, classification of the body part shown, and the jointly classification task. On this basis, the architecture developed, first based on CLIP and then redefined with Flamingo, obtained satisfactory results, thanks to the usage of an in-context learning approach, in which an adaptive few-shot paradigm was used in a RAG architecture.

Promising results were further stressed for a subsequent experimental phase, in which the model was asked to provide a diagnostic suggestion. Here, the retrieval approach involved an automatic created KG, instead of a vector database: created connections, leveraging the entire documentary corpus, allowed a broader retrieval that does not only rely on a single document, but across the information contained in different ones.

The main application differences can be found between the two considered domains, namely the educational and medical domain. However, for the educational domain, which is not critical, the eventual trade-off with the use of closed-source is mainly linked to the usage of paid APIs. In contrast, applications that involve the interaction with sensitive information should not use external models to prevent privacy-related issues and data leakage. This is the case in the medical domain, for which local and open-source models are preferred and for which a suitable fine-tuning procedure, also involving prompt engineering strategies, may be highly beneficial.

In conclusion, the presented work showed how the generative capabilities of LLMs can be enriched not only with fine tuning, but also with prompt engineering and RAG architecture. Encoder-only and encoder-decoder models dominated NLP tasks since the introduction of the Attention Mechanism: decoder-only models had been shown to be an interesting and valid alternative to them for some traditional tasks such as semantic classification, QA and IR, surpassing encoder-only models, thus assessing a paradigm shift in NLP research.

However, decoder-only models require enormous computational resources for both fine-tuning and inference and an appropriate quantity of data. A second

paradigm shift is towards a new conception of fine-tuning, which can be revisited as injection of a well-structured knowledge, such as KG derived from domain-specific ontologies, or in an hyper-tuning over generation parameter, or in enhancing prompt engineering techniques.

6.1 Future works

The results and related conclusions obtained would benefit from further theoretical considerations and the experimental phase, thus making those future works a natural continuation of the research presented in this doctoral dissertation. Some of those future directions, presented below, are currently under investigation and involve stressing the potentialities of diverse models and approaches that were not explored in the context of the PhD years.

Although it is clear that, at least for a certain typology of tasks, the usage of traditional techniques, LMs and encoder-only models are outdated, the idea to how use novels LLMs for traditional tasks is spreading and gaining both popularity, scientific interest, and outstanding performances. LMs has been a valuable models for the majority NLP tasks, but those can be easily performed via LLMs and *ad hoc* designed prompt.

The results obtained in the RE task (RQ1) lays in this context: they were obtained when the general interest of the scientific community started to shift toward generative models. This caused a redefinition of the scientific context and of the RQs reported in this work.

Since the proposed approaches did not establish a significant improvement in SemEval 2010 Task 8, and the same data set can be seen to be almost saturated, two diverse data sets were considered more challenging, since both involve document-level RE. The selected data sets are DocRED [146] and the E3C corpus [84]: the former is a general data set while the latter focuses on clinical reports.

For both data sets, an approach involving generative LLM in context of traditional NLP tasks such as RE, is currently being investigated. In particular, a RAG-based setup is under definition, in which the retrieve component can be performed on a well-structured knowledge base, such as a KG defined over an ontology of the domain considered. Retrieved information can serve as samples in a few-shot scenario, but also as a context to produce the expected output.

Related works, also including the development of effective decoding strategies, crucial for a correct evaluation of the generated text by LLM, represent the future direction of this work, along with a comparison between techniques involving encoder-only and decoder-only models or a joint usage of both models.

As for the Unipa-GPT framework, a further experimental phase over diverse LLMs is required to explore valuable alternative to GPT models which outperform in both chatbot and simple QA configuration. In this direction, an hyper-tuning over generation parameter, along with different fine-tuning strategies, may be worth exploring for future related works.

In addition, studies involving newer retrieval techniques and a different knowledge arrangement may greatly benefit the overall framework, along with the possibility of adding newer documents, thus allowing for more complex questions. In this sense, a well-structured definition of a retrieval phase in a KG structure, would be beneficial for the entire framework, thus allowing for a more grounded retrieval phase. The definition of a suitable arrangement of documents in a graph-like structure and a proper retrieval strategy are under investigation.

In addition, the UniQA data set can benefit from a comprehensive analysis of performances of different models in both Italian and English, as well as a more grounded retrieval strategy. In this context, this data set is a valuable source thanks to the annotated golden context and answer.

In the DR-LLM framework, a KG structure was obtained leveraging a RE-oriented prompt over an unstructured data set. Despite the promising results obtained, it can be beneficial to rely on a well-defined strategy for building the KG, with the addition of domain information. The idea is to use an external validated ontology by domain expert, as a guide to properly map the knowledge in the data so that a grounded KG can be obtained.

In addition, a more comprehensive evaluation of the generated text, which involves domain experts, would be beneficial for the entire system and would also provide valuable suggestions for future work.

The general idea towards well-structured external data sources, arranged in KGs or ontologies of the given domain, and the development of a proper retrieval strategy, are the main directions and the natural prosecution of this PhD thesis, and they are currently under investigation.

Chapter 7

Appendix A

Translated tables from Chapter 4.

Table 7.1: English version of Table 4.4.

instruction type	instruction text
custom instruction	You are Unipa-GPT, the chatbot and virtual assistant of the University of Palermo. Provide a cordially and colloquially answers to the questions provided. If you receive a greeting, answer by greeting and introducing yourself. If you receive a question concerning the University of Palermo, answer relying on the documents given to you with the question. If you do not know how to answer, apologise and suggest that you consult the website, do not invent answers. Question: {question} Documents: {context}
condensed instruction	Given the following conversation and question, rephrase the next question in such a way that it is a single question. Conversation: {chat.history} Next question: {question} Single question:

Table 7.2: English version of Table 4.6.

<i>condensed-emb</i>	<i>condensed-FT-emb</i>
<i>what the all-inclusive fee is</i>	
The all-inclusive fee for enrolment at the University of Palermo for the academic year 2023/24 are calculated according to the student's ISEE income class and the benefits provided as part of the right to study university. The payment of fees and the all-inclusive fee is divided into three instalments, the first of which must be paid by 30 September 2023 for enrolment in open access study courses and for enrolment to years subsequent to the first. To find out the terms and amounts of payment of fees and the all-inclusive fee, we advise you to consult the Teaching Calendar and the Calendar and the Student Contribution Regulations in force. In case of any doubts or specific questions, contact the Student secretariat directly of the University of Palermo.	The all-inclusive fee is the payment of the tuition fees that students have to pay at the time of university enrolment or registration. Payment is divided into three instalments: the first instalment, payable during the registration or enrolment, is the 40% of the amount due; the second instalment, to be paid by 29 December 2023, is the 30% of the amount due; the third instalment, to be paid by 29 February 2024, is the 30% of the amount due. Students enrolling for the first time to a three-year degree or a single-cycle master's degree, who graduated with a grade between 95 and 100, benefit, for the first year, from a reduction of 20% up to a maximum of €350.00. Those who graduate with a grade of 100 cum laude receive a 50% reduction for the first year, up to a maximum of a maximum of 500.00 euro. [...]

Table 7.3: English version of Table 4.7.

<i>custom-full</i>	<i>custom-FT-clear</i>
<i>is the bachelor degree course in mechanical engineering open or restricted access?</i>	
The bachelor degree course in mechanical engineering at the University of Palermo is open-access, with a number of places in local programming equal to 200. This is indicated in the document for the academic year 2023/2024 for the mechanics curriculum. If you need further information, I suggest you to consult the university website	The bachelor degree course in mechanical engineering is open-access
<i>do i have to take a test to enter the bachelor degree course in mechanical engineering?</i>	
Yes, to enter the bachelor degree course in mechanical engineering at the University of Palermo, you must take an admission test. The University's Guidance and Tutoring Center (COT) organises various initiatives to help you cope with entrance tests and pass them successfully, such as the 'How to cope with the entrance test' workshops and mock entrance tests. In addition, preparation courses are available for the admission tests, which will start on 17 July 2023 and will take place exclusively online on the Microsoft Teams platform. For more information, I advise you to consult the documents provided with your application.	Yes, there are tests for entry to the bachelor degree course in Mechanical Engineering.

Table 7.4: English version of Table 4.8

<i>condensed-clear</i>	<i>custom-FT-clear</i>
<i>which three-year course do you recommend I attend to then go on to the master's degree in aerospace engineering?</i>	
To move on to the master's degree in Aerospace Engineering at the University of Palermo, the recommended three-year course is the Bachelor's degree in Mechanical Engineering with Aeronautical curriculum. I advise you to consult the study plan for the three-year degree course and contact the Department of Engineering for further information.	I recommend that you attend the three-year degree course in Mechanical Engineering and then move on to the master's degree course in Aerospace Engineering. The three-year course will provide you with the necessary foundations for the master's degree.

Table 7.5: English version of Table 4.9.

condensed-emb	condensed-full	pure fine-tuned model
<i>first-year medicine course exam list</i>		
The exams for the first year of the medicine course at the University of Palermo for the academic year 2023/24 are not specified in the documents provided. I suggest you consult the call for applications for more information. Furthermore, as far as enrolment is concerned, open access courses can be enrolled without late fees from 1 August [...]	I am sorry, but I have not found any information regarding the first-year exams of the medicine course at the University of Palermo for the academic year 2023/24. I advise you to consult the call for applications for more information.	The list of exams for the first year of the degree course in Medicine at the University of Palermo includes the following exams: Human Anatomy with Exercises, Applied Biology, English, Biological Chemistry with Elements of Molecular Biology, Physics, Mathematics, Statistics and Computer Science, Human Histology with Elements of Cell Biology, Human Physiology with Exercises, and General Pathology.
<i>what are the subjects of the first year of medicine</i>		
The first-year subjects of the medical course at the University of Palermo for the 2023/24 academic year are: Physics I, Mathematics I and General and Organic Chemistry C.I. Instead, for the degree course in Biological Sciences, the first-year subjects are: Mathematics with Exercises, Cytology and Histology with Exercises, General Chemistry with Exercises and General Botany and Histology with Exercises.	The first-year subjects of the medical course at the University of Palermo for the academic year 2023/24 are: Physics I, Mathematics I, General and Organic Chemistry C.I., Organic Chemistry, Biology, Embryology and Histology C.I., Genetics and Microbiology C.I., Computer Science, Bioinformatics and Technical Scientific English C.I. I advise you to consult the documents provided for more information.	The first-year medical subjects are: Human Anatomy, Histology, Embryology, Biochemistry, Cell Biology, Genetics, Physics, Chemistry, English, Methodology of Intellectual Work, and Culture and Values in Healthcare.

Table 7.6: English version of Table 4.10.

instruction type	instruction text
<i>custom instruction</i>	I am Unipa-GPT, a chatbot and virtual assistant of the University of Palermo, who responds cordially and in a colloquial manner. To greetings, answer by greeting and introducing yourself; Answer the question with the words “Answer: ” Remember that the rector of the university is Professor Massimo Midiri. If the question concerns the University of Palermo, answer on the basis of the information and provide the links associated with it; If you do not know how to answer the question, answer by saying that you are an artificial intelligence that still has a lot to learn and suggest that you go to https://www.unipa.it/, do not invent answers. Question: {question} Documents: {context}
<i>condensed instruction</i>	Given the following conversation and question, rephrase the next question in such a way that it is a single question. Conversation: {chat_history} Next question: {question} Single question:

Table 7.7: List of the six QA pairs used for evaluation purposes

ID	Question	Answer
Q1	<i>Chi è il professore di Intelligenza Artificiale 1 per il corso di Laurea Magistrale in Ingegneria Informatica?</i>	Il professore di Intelligenza Artificiale 1 del corso di Laurea Magistrale Magistrale in Ingegneria Informatica è il professore Gaglio e verrà erogata durante il primo semestre. Per maggiori informazioni vai su http://www.unipa.it/struttura.html?id=721
Q2	<i>Quali sono le scadenze di iscrizione ad un corso di laurea?</i>	La presentazione delle domande di iscrizione per l'Anno Accademico 2023/2024 varia in base alla tipologia di corso. Per i corsi di studio triennali e magistrali a ciclo unico, la domanda può essere presentata dall'1 agosto al 30 settembre 2023, mentre per i corsi magistrali, dall'1 agosto al 30 novembre 2023, in entrambi i casi, è richiesto il pagamento della prima rata delle tasse universitarie. Per i corsi di studio ad accesso programmato locale, con prenotazione online o ad accesso programmato nazionale, le immatricolazioni vanno effettuate entro i termini previsti dai relativi bandi di concorso.
Q3	<i>Come funziona la magistrale di chimica?</i>	Il corso di laurea magistrale in Chimica presso l'Università degli Studi di Palermo, ha sede a Palermo e ha una durata di 2 anni. L'obiettivo del corso è la formazione di laureati con una solida preparazione di base, che li ponga in grado di affrontare con competenza ed autonomia sia attività professionali sia quelle di ricerca accademica ed industriale. Per questo motivo il corso di studio è progettato con lo scopo di completare la cultura scientifica nelle discipline chimiche fondamentali e di introdurre conoscenze più avanzate. Quindi lo studente può completare il proprio piano di studi, attenzionando gli insegnamenti che meglio si adattano ai propri interessi specifici, come la chimica dei materiali, dell'ambiente e dei beni culturali, la chimica supramolecolare e la didattica della chimica. Per maggiori informazioni vai su http://www.unipa.it/struttura.html?id=766
Q4	<i>ciao! sono un ragazzo appena uscito dal liceo che è interessato al settore legale, in particolare alle leggi sulle aziende. dove potrei iscrivermi?</i>	Ciao! Il corso di Laurea in Consulente Giuridico d'Impresa sembra proprio fare al caso tuo! Il corso di laurea ha sede a Trapani e ha una durata di 3 anni e ti fornirà le conoscenze in ambito giuridico ed economico-aziendalistico. Una volta terminato il percorso di studio potrai svolgere attività interdisciplinari che richiedono competenze giuridiche, aziendalistiche e organizzativo-gestionali. Per maggiori informazioni vai su http://www.unipa.it/struttura.html?id=1557
Q5	<i>come posso prenotare un appuntamento in segreteria?</i>	È possibile recarsi in segreteria il lunedì, mercoledì e venerdì dalle 10.00 alle 12.00, martedì e giovedì dalle 15.00 alle 17.00. Puoi prenotare il tuo turno attraverso la App SolariQ. Per maggiori informazioni vai su https://www.unipa.it/servizi/segreterie/
Q6	<i>Come si pagano le tasse?</i>	Il pagamento delle tasse deve essere effettuato esclusivamente mediante sistema PAgoPA (Pagamenti della Pubblica Amministrazione) al quale si accede tramite portale di ateneo. Dopo aver compilato la pratica online, è possibile pagare direttamente online con il sistema PAgoPA o stampare il bollettino e pagare presso una ricevitoria abilitata PAgoPA. Ulteriori informazioni sul pagamento via PAgoPA sono reperibili qui https://immaweb.unipa.it/immaweb/public/pagamenti.seam , mentre è disponibile il Regolamento in materia di contribuzione studentesca https://www.unipa.it/servizi/segreterie/.content/documenti/regolamenti_calendari/2023/5105144-def_regolamento-contribuzione-studentesca-2023-24-2.pdf

Table 7.8: English version of Table 7.7.

ID	Questions	Answers
Q1	<i>Who is the Artificial Intelligence 1 professor for Computer Engineering Master degree course?</i>	The Artificial Intelligence 1 professor for the Computer Engineering Master degree course is Professor Gaglio and it will be delivered during the first semester. For more information go to http://www.unipa.it/struttura.html?id=721
Q2	<i>What are the deadlines for enrolling in a degree programme?</i>	The submission of applications for the Academic Year 2023/2024 varies according to the type of course. For three-year and single-cycle master's degree courses, applications can be submitted from 1 August to 30 September 2023, while for master's degree courses, from 1 August to 30 November 2023; in both cases, payment of the first instalment of tuition fees is required. For courses with local programmed access, with online booking or national programmed access, enrolment must be carried out by the deadlines set out in the corresponding calls for application.
Q3	<i>How does the master's degree in chemistry work?</i>	The Master's degree course in Chemistry at the University of Palermo is based in Palermo and lasts 2 years. The aim of the course is to train graduates with a good background, enabling them to deal competently and independently with both professional activities and academic and industrial research. For this reason, the course is designed with the aim of completing the scientific culture in the fundamental chemical disciplines and introducing more advanced knowledge. Therefore, students can complete their study plan by focusing on the subjects that best suit their specific interests, such as the chemistry of materials, the environment and cultural heritage, supramolecular chemistry and the didactics of chemistry. For more information go to http://www.unipa.it/struttura.html?id=766
Q4	<i>hello! I'm a guy just out of high school who is interested in law, especially corporate law. where should i apply?</i>	Hi! The Bachelor of Business Law Consultant programme sounds like it could be just the thing for you! The degree course is based in Trapani and lasts 3 years and will provide you with knowledge in the fields of law and business economics. Once you have completed the course you will be able to carry out interdisciplinary activities requiring legal, business and organisational-managerial skills. For more information go to http://www.unipa.it/struttura.html?id=1557
Q5	<i>how can i book an appointment at the secretariat?</i>	You can go to the secretariat on Mondays, Wednesdays and Fridays from 10 a.m. to 12 noon, Tuesdays and Thursdays from 3 p.m. to 5 p.m. . You can book your appointment through the SolariQ App. For more information go to https://www.unipa.it/servizi/segreteria/
Q6	<i>How do I pay fees?</i>	Fees must be paid exclusively through the PAgoPA (Public Administration Payments) system, which is accessed through the university portal. After completing the paperwork online, you can either pay directly online via the PAgoPA system or print out the payment slip and pay at a PAgoPA-enabled tax office. Further information on paying via PAgoPA can be found here https://immaweb.unipa.it/immaweb/public/pagamenti.seam , while the Student Contribution Regulations is available here https://www.unipa.it/servizi/segreteria/.content/documents/regulations_calendars/2023/5105144-def_regulation-student-contribution-2023-24-2.pdf

Bibliography

- [1] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan. Flamingo: a Visual Language Model for Few-Shot Learning. *arXiv preprint arXiv:2204.14198*, 2022.
- [2] T. Alkire and C. Rosen. *Romance languages: A historical introduction*. Cambridge University Press, 2010.
- [3] Anthropic. Claude. <https://www.anthropic.com/news/introducing-claude>.
- [4] A. Awadalla, I. Gao, J. Gardner, J. Hessel, Y. Hanafy, W. Zhu, K. Marathe, Y. Bitton, S. Gadre, S. Sagawa, J. Jitsev, S. Kornblith, P. W. Koh, G. Ilharco, M. Wortsman, and L. Schmidt. OpenFlamingo: An Open-Source Framework for Training Large Autoregressive Vision-Language Models. *arXiv preprint arXiv:2308.01390*, 2023.
- [5] A. Bacciu, G. Trappolini, A. Santilli, E. Rodolà, and F. Silvestri. Fauno: The Italian Large Language Model that will leave you senza parole! *arXiv preprint arXiv:2306.14457*, 2023.
- [6] L. Baldini Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski. Matching the Blanks: Distributional Similarity for Relation Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.
- [7] S. Banerjee and A. Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005.

- [8] P. Basile, E. Musacchio, M. Polignano, L. Siciliani, G. Fiameni, and G. Semeraro. LLaMAntino: LLaMA 2 Models for Effective Text Generation in Italian Language. *arXiv preprint arXiv:2312.09993*, 2023.
- [9] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. Rangel Pardo, P. Rosso, and M. Sanguinetti. SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, 2019.
- [10] G. Berruto. Variazione diafasica, 2011. [https://www.treccani.it/enciclopedia/variazione-diafasica_\(Enciclopedia-dell'Italiano\)](https://www.treccani.it/enciclopedia/variazione-diafasica_(Enciclopedia-dell'Italiano)).
- [11] V. Bhat, S. D. Cheerla, J. R. Mathew, N. Pathak, G. Liu, and J. Gao. Retrieval Augmented Generation (RAG) Based Restaurant Chatbot with AI Testability. In *2024 IEEE 10th International Conference on Big Data Computing Service and Machine Learning Applications (BigDataService)*, 2024.
- [12] T. Boccato, M. Ferrante, and N. Toschi. Two-Phase RAG-Based Chatbot for Italian Funding Application Assistance, 2024.
- [13] C. Bosco, D. Felice, F. Poletto, M. Sanguinetti, T. Maurizio, et al. Overview of the EVALITA 2018 Hate Speech Detection Task. In *Ceur workshop proceedings*, 2018.
- [14] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language Models are Few-Shot Learners. *Advances in neural information processing systems*, 2020.
- [15] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998.
- [16] F. Celli, M. Lai, A. Duzha, C. Bosco, and V. Patti. Polycorpus XL: An Italian Corpus for the Detection of Hate Speech Against Politics. In *CLiC-it*, 2021.
- [17] S. Chan and A. Fyshe. Social and Emotional Correlates of Capitalization on Twitter. In *Proceedings of the Second Workshop on Computational Modeling of People’s Opinions, Personality, and Emotions in Social Media*, June 2018.

- [18] H. A. H. Chaudhry, R. Renzulli, D. Perlo, F. Santinelli, S. Tibaldi, C. Cristiano, M. Grosso, G. Limerutti, A. Fiandrotti, M. Grangetto, et al. Uniochest: A lung image dataset for segmentation of cancerous nodules on ct scans. In *International Conference on Image Analysis and Processing*, 2022.
- [19] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. *arXiv preprint arXiv:2402.03216*, 2024.
- [20] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. PaLM: Scaling Language Modeling with Pathways. *arXiv preprint arXiv:2204.02311*, 2022.
- [21] Chroma. <https://www.trychroma.com/>.
- [22] K. W. Church. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, 1988.
- [23] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov. Unsupervised Cross-lingual Representation Learning at Scale, 2019.
- [24] D. Croce, A. Zelenanska, and R. Basili. Neural Learning for Question Answering in Italian. In C. Ghidini, B. Magnini, A. Passerini, and P. Traverso, editors, *AI*IA 2018 – Advances in Artificial Intelligence*, 2018.
- [25] D. Demner-Fushman, M. D. Kohli, M. B. Rosenman, S. E. Shooshan, L. Rodriguez, S. Antani, G. R. Thoma, and C. J. McDonald. Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association*, 2016.
- [26] Detmeters and Tim and Artidoro Pagnoni and Ari Holtzman and Luke Zettlemoyer. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv preprint arXiv:2305.14314*, 2023.
- [27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *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)*, 2019.

- [28] Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, B. Chang, X. Sun, L. Li, and Z. Sui. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.
- [29] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou. The Faiss library. *arXiv preprint arXiv:2401.08281*, 2024.
- [30] S. D’Urso and F. Sciarrone. AI4LA: An Intelligent Chatbot for Supporting Students with Dyslexia, Based on Generative AI. In A. Sifaleras and F. Lin, editors, *Generative Intelligence and Intelligent Tutoring Systems*, 2024.
- [31] S. Es, J. James, L. Espinosa Anke, and S. Schockaert. RAGAs: Automated Evaluation of Retrieval Augmented Generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, 2024.
- [32] E. Fersini, F. Gasparini, G. Rizzi, A. Saibene, B. Chulvi, P. Rosso, A. Lees, and J. Sorensen. SemEval-2022 Task 5: Multimedia Automatic Misogyny Identification. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, 2022.
- [33] R. Figliè, T. Turchi, G. Baldi, and D. Mazzei. Towards an LLM-based Intelligent Assistant for Industry 5.0. In *Proceedings of the 1st International Workshop on Designing and Building Hybrid Human–AI Systems (SYNERGY 2024)*, 2024.
- [34] A. Forootani, D. E. Aliabadi, and D. Thraen. Bio-Eng-LMM AI Assist chatbot: A Comprehensive Tool for Research and Education. *arXiv preprint arXiv:2409.07110*, 2024.
- [35] U. Gava, F. D’Agata, E. Bennink, E. Tartaglione, D. Perlo, A. Vernone, F. Bertolino, E. Ficiarà, A. Cicerale, F. Pizzagalli, C. Guiot, M. Grangetto, and M. Bergui. UniTOBrain, 2021.
- [36] GeminiTeam, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, et al. Gemini: A Family of Highly Capable Multimodal Models, 2024.
- [37] GEMiniTeam, G. Comanici, E. Bieber, M. Schaekermann, I. Pasupat, N. Sachdeva, I. Dhillon, M. Blistein, O. Ram, D. Zhang, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025.

- [38] X. Geng, A. Gudibande, H. Liu, E. Wallace, P. Abbeel, S. Levine, and D. Song. Koala: A Dialogue Model for Academic Research, 2023. <https://bair.berkeley.edu/blog/2023/04/03/koala/>.
- [39] S. Ghanbari Haez, M. Segala, P. Bellan, S. Magnolini, L. Sanna, M. Consolandi, and M. Dragoni. A Retrieval-Augmented Generation Strategy to Enhance Medical Chatbot Reliability. In *Artificial Intelligence in Medicine*, 2024.
- [40] E. Grave, P. Bojanowski, P. Gupta, A. Joulin, and T. Mikolov. Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.
- [41] Z. Guo, Y. Zhang, and W. Lu. Attention Guided Graph Convolutional Networks for Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.
- [42] I. Hartsock and G. Rasool. Vision-language models for medical report generation and visual question answering: A review. *arXiv preprint arXiv:2403.02469*, 2024.
- [43] X. He, Y. Zhang, L. Mou, E. Xing, and P. Xie. Pathvqa: 30000+ questions for medical visual question answering. *arXiv preprint arXiv:2003.10286*, 2020.
- [44] M. Heath. Orthography in social media: Pragmatic and prosodic interpretations of caps lock. *Proceedings of the Linguistic Society of America*, 2018.
- [45] I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, 2010.
- [46] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 11 1997.
- [47] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. The Curious Case of Neural Text Degeneration. In *International Conference on Learning Representations*, 2020.
- [48] C. D. Hromei, D. Croce, V. Basile, and R. Basili. ExtremITA at EVALITA 2023: Multi-Task Sustainable Scaling to Large Language Models at its Extreme. In *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (2023)*, 2023.

- [49] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*, 2022.
- [50] S.-C. Huang, Z. Huo, E. Steinberg, C.-C. Chiang, M. P. Lungren, C. P. Langlotz, S. Yeung, N. H. Shah, and J. A. Fries. INSPECT: a multimodal dataset for pulmonary embolism diagnosis and prognosis. *arXiv preprint arXiv:2311.10798*, 2023.
- [51] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.
- [52] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung. Survey of Hallucination in Natural Language Generation. *ACM Computing Survey*, 2023.
- [53] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.
- [54] H. Jiang, Q. Bao, Q. Cheng, D. Yang, L. Wang, and Y. Xiao. Complex Relation Extraction: Challenges and Opportunities, 2020.
- [55] A. E. Johnson, T. J. Pollard, S. J. Berkowitz, N. R. Greenbaum, M. P. Lungren, C.-y. Deng, R. G. Mark, and S. Horng. MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports. *Scientific data*, 2019.
- [56] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy. SpanBERT: Improving Pre-training by Representing and Predicting Spans, 2020.
- [57] K. Junseong, L. Seolhwa, K. Jihoon, G. Sangmo, K. Yejin, C. Minkyung, S. Jy-yong, and C. Chanyeol. Linq-Embed-Mistral: Elevating Text Retrieval with Improved GPT Data Through Task-Specific Control and Quality Refinement, 2024. <https://getlinq.com/blog/linq-embed-mistral/>.
- [58] D. Jurafsky and J. H. Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, 2025.

- [59] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [60] T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, 2017.
- [61] H. R. Kirk, W. Yin, B. Vidgen, and P. Röttger. SemEval-2023 Task 10: Explainable Detection of Online Sexism. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, 2023.
- [62] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in neural information processing systems*, 2012.
- [63] M. Kulkarni, P. Tangarajan, K. Kim, and A. Trivedi. Reinforcement Learning for Optimizing RAG for Domain Chatbots. *arXiv preprint arXiv:2401.06800*, 2024.
- [64] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001.
- [65] M. Lai, F. Celli, A. Ramponi, S. Tonelli, C. Bosco, and V. Patti. HaSpeede3 at EVALITA 2023: Overview of the Political and Religious Hate Speech Detection task. In *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023)*, 2023.
- [66] M. Lai, S. Menini, M. Polignano, V. Russo, R. Sprugnoli, G. Venturi, et al. Evalita 2023: Overview of the 8th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. In *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023)*, 2023.
- [67] J. J. Lau, S. Gayen, A. Ben Abacha, and D. Demner-Fushman. A dataset of clinically generated visual questions and answers about radiology images. *Scientific data*, 2018.
- [68] Y. LeCun. Generalization and network design strategies. *Connections in Perspective*, 1989.

- [69] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 2019.
- [70] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*, 2020.
- [71] C. Li and Y. Tian. Downstream Model Design of Pre-trained Language Model for Relation Extraction Task, 2020.
- [72] C. Li, C. Wong, S. Zhang, N. Usuyama, H. Liu, J. Yang, T. Naumann, H. Poon, and J. Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36, 2024.
- [73] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, and M. Zhang. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*, 2023.
- [74] V. Lialin, V. Deshpande, X. Yao, and A. Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint 2303.15647*, 2024.
- [75] C.-Y. Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 2004.
- [76] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, 2014.
- [77] W. Lin, Z. Zhao, X. Zhang, C. Wu, Y. Zhang, Y. Wang, and W. Xie. Pmc-clip: Contrastive language-image pre-training using biomedical documents. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2023.
- [78] B. Liu, L.-M. Zhan, L. Xu, L. Ma, Y. Yang, and X.-M. Wu. Slake: A semantically-labeled knowledge-enhanced dataset for medical visual question answering. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, 2021.

- [79] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.
- [80] J. Liu. LlamaIndex, 2022. https://github.com/jerryjliu/llama_index.
- [81] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach, 2019.
- [82] LlamaTeam. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024.
- [83] C. K. Lo. What Is the Impact of ChatGPT on Education? A Rapid Review of the Literature. *Education Sciences*, 2023.
- [84] B. Magnini, B. Altuna, A. Lavelli, M. Speranza, and R. Zanoli. The E3C Project: Collection and Annotation of a Multilingual Corpus of Clinical Cases. *Proceedings of the Seventh Italian Conference on Computational Linguistics*, 2020.
- [85] B. Magnini, B. Altuna, A. Lavelli, M. Speranza, and R. Zanoli. The E3C Project: Collection and Annotation of a Multilingual Corpus of Clinical Cases. *Proceedings of the Seventh Italian Conference on Computational Linguistics CLiC-it 2020*, 2020.
- [86] A. Mandya, D. Bollegala, and F. Coenen. Graph Convolution over Multiple Dependency Sub-graphs for Relation Extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, 2020.
- [87] J. A. Meaney, S. Wilson, L. Chiruzzo, A. Lopez, and W. Magdy. SemEval 2021 Task 7: HaHackathon, Detecting and Rating Humor and Offense. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, 2021.
- [88] MedPix. <https://medpix.nlm.nih.gov/home>.
- [89] S. Menini, R. Sprugnoli, and A. Uva. “Who was Pietro Badoglio?” Towards a QA system for Italian History. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, 2016.
- [90] T. Mikolov. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*, 2013.

- [91] M. Moor, Q. Huang, S. Wu, M. Yasunaga, Y. Dalmia, J. Leskovec, C. Zakka, E. P. Reis, and P. Rajpurkar. Med-flamingo: a multimodal medical few-shot learner. In *Machine Learning for Health (ML4H)*, pages 353–367. PMLR, 2023.
- [92] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 2023.
- [93] T. Nayak, N. Majumder, P. Goyal, and S. Poria. Deep Neural Approaches to Relation Triplets Extraction: a Comprehensive Survey . *Cognitive Computation*, 2021.
- [94] E. Nossem. Queer, Frocia, Femminiellø, Ricchione et al.–Localizing “Queer” in the Italian Context. *gender/sexuality/italy*, 2019.
- [95] D. Nozza, A. T. Cignarella, G. Damo, T. Caselli, and V. Patti. HODI at EVALITA 2023: Overview of the Homotransphobia Detection in Italian Task. In *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023)*, 2023.
- [96] D. Nozza and D. Hovy. The State of Profanity Obfuscation in Natural Language Processing Scientific Publications. In *Findings of the Association for Computational Linguistics: ACL 2023*, 2023.
- [97] OpenAI. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*, 2024.
- [98] R. Orlando, L. Moroni, P.-L. H. Cabot, E. Barba, S. Conia, S. Orlandini, G. Fiameni, and R. Navigli. Minerva LLMs: The First Family of Large Language Models Trained from Scratch on Italian Data. *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, 2024.
- [99] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.
- [100] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002.
- [101] O. Pelka, S. Koitka, J. Rückert, F. Nensa, and C. M. Friedrich. Radiology objects in context (roco): a multimodal image dataset. In *Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of*

Biomedical Data and Expert Label Synthesis: 7th Joint International Workshop, CVII-STENT 2018 and Third International Workshop, LABELS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings 3, 2018.

- [102] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014.
- [103] Pinecone. <https://www.pinecone.io/>.
- [104] M. Polignano, P. Basile, M. de Gemmis, G. Semeraro, and V. Basile. ALBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets. In *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019)*. CEUR, 2019.
- [105] M. Polignano, P. Basile, and G. Semeraro. Advanced Natural-based interaction for the ITALian language: LLaMAntino-3-ANITA. *arXiv preprint arXiv:2405.07101*, 2024.
- [106] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning Transferable Visual Models from Natural Language Supervision. In *International conference on machine learning*, 2021.
- [107] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving Language Understanding by Generative Pre-Training, 2018.
- [108] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language Models are Unsupervised Multitask Learners. *OpenAI blog*, 2019.
- [109] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 2020.
- [110] M. M. Rahman and Y. Watanobe. ChatGPT for Education and Research: Opportunities, Threats, and Strategies. *Applied Sciences*, 2023.
- [111] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv preprint arXiv:1606.05250*, 2016.
- [112] R. Rodriguez-Torrealba, E. Garcia-Lopez, and A. Garcia-Cabot. End-to-End generation of Multiple-Choice questions using Text-to-Text transfer Transformer models. *Expert Systems with Applications*, 2022.

- [113] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
- [114] M. Sanguinetti, G. Comandini, E. D. Nuovo, S. Frenda, M. A. Stranisci, C. Bosco, T. Caselli, V. Patti, and I. Russo. HaSpeeDe 2 @ EVALITA2020: Overview of the EVALITA 2020 Hate Speech Detection Task. *EVALITA Evaluation of NLP and Speech Tools for Italian - December 17th, 2020*, 2020.
- [115] A. Santilli and E. Rodolà. Camoscio: an Italian Instruction-tuned LLaMA. *arXiv preprint arXiv:2307.16456*, 2023.
- [116] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, 2009.
- [117] S. Schulz, J. Ševa, S. Rodriguez, M. Ostendorff, and G. Rehm. Named Entities in Medical Case Reports: Corpus and Experiments. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020.
- [118] S. Schweter. Italian BERT and ELECTRA models, 2020. <https://doi.org/10.5281/zenodo.4263142>.
- [119] I. Siragusa, S. Contino, M. L. Ciura, R. Alicata, and R. Pirrone. MedPix 2.0: A Comprehensive Multimodal Biomedical Dataset for Advanced AI Applications. In *Proceedings of the 3rd Italian Conference on Big Data and Data Science, ITADATA2024*, 2024.
- [120] I. Siragusa, S. Contino, M. L. Ciura, R. Alicata, and R. Pirrone. MedPix 2.0: A Comprehensive Multimodal Biomedical Data Set for Advanced AI Applications with Retrieval Augmented Generation and Knowledge Graphs. *Data Science and Engineering*, 2025.
- [121] I. Siragusa, S. Contino, and R. Pirrone. DR-Minerva: A Multimodal Language Model Based on Minerva for Diagnostic Information Retrieval. In *AIxIA 2024 – Advances in Artificial Intelligence*, 2025.
- [122] I. Siragusa and R. Pirrone. CHILab at HODI: A minimalist approach. In *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023)*, 2023.
- [123] I. Siragusa and R. Pirrone. Conditioning Chat-GPT for Information Retrieval: The Unipa-GPT Case Study. In *Proceedings of the Seventh Workshop on Natural Language for Artificial Intelligence (NL4AI 2023) co-located*

with 22th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2023), 2023.

- [124] I. Siragusa and R. Pirrone. Unipa-GPT: A Framework to Assess Open-source Alternatives to Chat-GPT for Italian Chat-bots. In *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, 2024.
- [125] I. Siragusa and R. Pirrone. Unipa-GPT: Large Language Models for university-oriented QA in Italian. *Italian Journal of Computational Linguistics*, 2024.
- [126] I. Siragusa and R. Pirrone. UniQA: an Italian and English Question-Answering Data Set Based on Educational Documents. In *Proceedings of the Eighth Workshop on Natural Language for Artificial Intelligence (NL4AI 2024) co-located with 23th International Conference of the Italian Association for Artificial Intelligence (AI*IA 2024)*, 2024.
- [127] I. Siragusa, R. Pirrone, et al. CHILab at HaSpeeDe3: Overview of the Taks A Textual. In *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023)*, 2023.
- [128] L. Song, Y. Zhang, D. Gildea, M. Yu, Z. Wang, and J. Su. Leveraging Dependency Forest for Neural Medical Relation Extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.
- [129] S. Subramanian, L. L. Wang, S. Mehta, B. Bogin, M. van Zuylen, S. Parasa, S. Singh, M. Gardner, and H. Hajishirzi. Mediat: A dataset of medical images, captions, and textual references. *arXiv preprint arXiv:2010.06000*, 2020.
- [130] A. Tack and C. Piech. The AI Teacher Test: Measuring the Pedagogical Ability of Blender and GPT-3 in Educational Dialogues. In *Proceedings of the 15th International Conference on Educational Data Mining*, Durham, United Kingdom, 2022.
- [131] Q. Tao, X. Luo, H. Wang, and R. Xu. Enhancing Relation Extraction Using Syntactic Indicators and Sentential Contexts . In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, 2019.

- [132] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto. Alpaca: A Strong, Replicable Instruction-Following Model, 2023. <https://crfm.stanford.edu/2023/03/13/alpaca.html>.
- [133] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*, 2023.
- [134] S. Vakayil, D. S. Juliet, A. J., and S. Vakayil. RAG-Based LLM Chatbot Using Llama-2. In *2024 7th International Conference on Devices, Circuits and Systems (ICDCS)*, 2024.
- [135] M.-H. Van, P. Verma, and X. Wu. On Large Visual Language Models for Medical Imaging Analysis: An Empirical Study. *arXiv preprint arXiv:2402.14162*, 2024.
- [136] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 2008.
- [137] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, 2017.
- [138] S. Vidivelli, M. Ramachandran, and A. Dharunbalaji. Efficiency-Driven Custom Chatbot Development: Unleashing LangChain, RAG, and Performance-Optimized LLM Fusion. *Computers, Materials & Continua*, 2024.
- [139] C. Wang, J. Ong, C. Wang, H. Ong, R. Cheng, and D. Ong. Potential for GPT technology to optimize future clinical decision-making using retrieval-augmented generation. *Annals of Biomedical Engineering*, 2024.
- [140] H. Wang, M. Tan, M. Yu, S. Chang, D. Wang, K. Xu, X. Guo, and S. Potdar. Extracting Multiple-Relations in One-Pass with Pre-Trained Transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.
- [141] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei. Improving Text Embeddings with Large Language Models. *arXiv preprint arXiv:2401.00368*, 2023.
- [142] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, and F. Wei. Multilingual E5 Text Embeddings: A Technical Report. *arXiv preprint arXiv:2402.05672*, 2024.

- [143] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.
- [144] J. Weizenbaum. ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 1983.
- [145] C. Xu, D. Guo, N. Duan, and J. McAuley. Baize: An open-source chat model with parameter-efficient tuning on self-chat data, 2023.
- [146] Y. Yao, D. Ye, P. Li, X. Han, Y. Lin, Z. Liu, Z. Liu, L. Huang, J. Zhou, and M. Sun. DocRED: A Large-Scale Document-Level Relation Extraction Dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.
- [147] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, 2019.
- [148] M. Zampieri, P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, and c. Çöltekin. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*, 2020.
- [149] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao. Relation Classification via Convolutional Deep Neural Network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014.
- [150] S. Zhang, Y. Xu, N. Usuyama, H. Xu, J. Bagga, R. Tinn, S. Preston, R. Rao, M. Wei, N. Valluri, et al. BiomedCLIP: a multimodal biomedical foundation model pretrained from fifteen million scientific image-text pairs. *arXiv preprint arXiv:2303.00915*, 2023.
- [151] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. BERTScore: Evaluating Text Generation with BERT. *arXiv preprint arXiv:1904.09675*, 2020.
- [152] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen, L. Wang, A. T. Luu, W. Bi, F. Shi, and S. Shi. Siren’s Song in the AI Ocean: A Survey on Hallucination in Large Language Models, 2023.

- [153] Y. Zhang, P. Qi, and C. D. Manning. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.
- [154] Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, 2017.
- [155] K. Zhao, H. Xu, Y. Cheng, X. Li, and K. Gao. Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction. *Knowledge-Based Systems*, 2021.
- [156] Y. Zhao, H. Wan, J. Gao, and Y. Lin. Improving Relation Classification by Entity Pair Graph. In *Proceedings of The Eleventh Asian Conference on Machine Learning*, 2019.
- [157] Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh. Calibrate Before Use: Improving Few-shot Performance of Language Models. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.
- [158] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.
- [159] H. Zhu, Y. Lin, Z. Liu, J. Fu, T.-S. Chua, and M. Sun. Graph Neural Networks with Generated Parameters for Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.