**ALMA MATER STUDIORUM**
**UNIVERSITÀ DI BOLOGNA**

**DEPARTMENT OF COMPUTER SCIENCE**
**AND ENGINEERING**

ARTIFICIAL INTELLIGENCE

**MASTER THESIS**

in

Natural Language Processing

# GRAPH-BASED APPROACHES FOR FEW-SHOT EXAMPLE SELECTION IN IN-CONTEXT LEARNING

CANDIDATE

Francesco Alfieri

SUPERVISOR

Prof. Andrea Galassi

CO-SUPERVISOR

Dr. Giulia Grundler

Academic year 2023-2024

Session 3rd

*In the current digitized world, trivial information is accumulating every second; preserved in all its triteness. Never fading, always accessible. Rumors about petty issues, misinterpretations, slander. All this junk data preserved in an unfiltered state, growing at an alarming rate. It will only slow down social progress, reduce the rate of evolution. The digital society furthers human flaws and selectively rewards development of convenient half-truths. Just look at the strange juxtapositions of morality around you. [...] You exercise your right to "freedom" and this is the result. All rhetoric to avoid conflict and protect each other from hurt. The untested truths spun by different interests continue to churn and accumulate in the sandbox of political correctness and value systems. Everyone withdraws into their own small gated community, afraid of a larger forum. They stay inside their little ponds leaking what ever "truth" suits them into the growing cesspool of society at large. The different cardinal truths neither clash nor mesh. No one is invalidated, but nobody is right. Not even natural selection can take place here. The world is being engulfed in "Truth". And this is the way the world ends. Not with a BANG, but a whimper.*

Metal Gear Solid 2 (2001)

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

In the last years, with the progressive and dramatic increase in size of Language Models, in addition to the traditional paradigm of pre-training and fine-tuning, the In-Context Learning approach has gained more and more popularity in a wide variety of tasks. While fine-tuning often allows to achieve better performances, the process is computationally expensive and requires large amounts of good-quality data. On the other hand, In-Context Learning allows to employ general-purpose Large Language Models on a wide range of tasks without the need for costly weight updates. While powerful and versatile, this method yields a new set of challenges that make for compelling research topics. One of such challenges is *prompt designing*. Specifically, a lot of studies are aimed at exploring effective ways to format prompt templates and organizing the demonstrations included in the prompt itself, with a focus on demonstration selection, reformatting and ordering.

In this thesis, we focus in particular on the demonstration selection aspect, for which we propose a fast, versatile and theoretically sound graph-based approach. The reference setting is one in which, given a query for a specific task, a set of demonstrations must be retrieved from an available large knowledge base. Several studies found that there exists a trade-off in effectiveness between selecting demonstrations that are relevant (usually meaning that they are *similar*) to the query and choosing examples that are semantically diverse from each other. Popular heuristic approaches include KNN-based [16] and greedy MMR-maximizing [28] methods. The goals of our method are to select examples that are both relevant to the query (via any suitable distance or similarity metric) and that cover a variety of concepts encoded in the knowledge base. Moreover, tools from graph theory allows to select examples that best summarize such concepts.

The proposed method can be applied in any task that can be formulated as described above. Here, we perform experiments evaluating several configurations of the retrieval approach on five distinct tasks from three different datasets, comparing them to Random and KNN-based baselines.

In Chapter 2 we introduce the theoretical concepts upon which our approach is grounded. In particular, Section 2.1 includes the relevant literature about In-Context Learning, which evidences the similarity-diversity tradeoff. Section 2.2 focuses on the description of the tools from graph theory that are used, with the associated motivations that support their adoption.

Chapter 3 is devoted to the full description of our proposed methodology. There we illustrate the process of creation of what we call the *Demonstration Graph*, a large graph where nodes represent demonstrations from the knowledge base and edges represent similarity relations between examples. At inference time, a subgraph is extracted depending on the query, and it is subsequently partitioned into dense and scarcely connected communities. Last, the most representative demonstrations from each community is selected and included in the prompt.

Chapter 4 includes several experiments aimed to test the effectiveness of our method. In Section 4.1 we tackle three tasks concerning argument mining in the legal domain. In Section 4.2 the goal is to assess the compliance to the GDPR of clauses from privacy policies of online companies in terms of comprehensiveness of information. Last, in Section 4.3 we test the capability of LLMs to learn to play the card game Chef's Hat by providing only examples.

In Chapter 5 we summarize our findings by highlighting both shortcomings and promising results. The approach we propose offers a huge design space that leaves room for possibly large improvements. Hence, we include in the same section several methods and topics that can be explored in order to further improve the performances in all tasks.

# Chapter 2

# Background

## 2.1 In-Context Learning

In recent years, the popularity of in-context learning (ICL) as a research topic has seen as significant rise, starting from the influential article *Language Models are Few-Shot Learners* [7], which accompanied the release of the GPT-3 transformer model. This paradigm consists in providing a task instruction to a large language model in natural language as a prompt, together with a number of demonstrations which act as examples. This approach has also been referred to as *zero-shot learning*, *one-shot learning*, or *few-shot learning*, depending on the number of provided demonstrations. One of the greatest appeals of in-context learning is that it allows large language models to tackle a variety of novel and complex tasks without the need for the expensive process of fine-tuning. It has been observed that the capability of successfully tackle this kind of tasks via in-context learning is an *emergent* ability of LLMs. In fact, by studying the performances of different models with similar sizes, it has been observed in [26] that for several tasks (such as Truthful Question Answering, 3-digit addition and subtraction and Word Unscrambling), the performances of models in the few-shot prompting setting are very close to random baselines until a certain threshold of number of parameters, after which they see a steep increase (Figure 2.1).

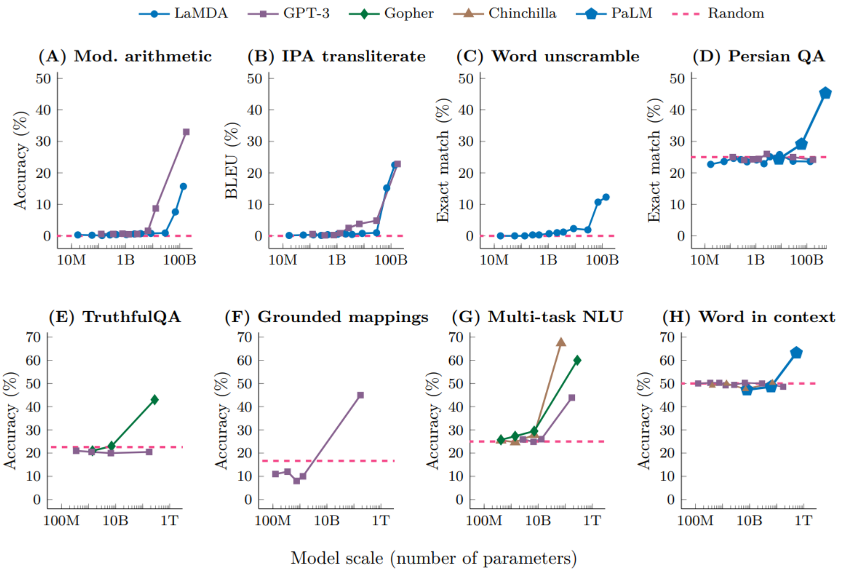

Figure 2.1: Examples of emergence in the few-shot prompting setting from [26].

Formally, if $D_k = \{f(x_1, y_1), \ldots, f(x_k, y_k)\}$ represents a set of demonstrations with $k$ examples and $f(x_k, y_k)$ is the prompt function that transforms the $k$-th task example into a natural language prompt, then the prediction of the output $\hat{y}_{k+1}$ generated from LLMs can be formulated as follows:

$$\text{LLM}\left(I, \underbrace{f(x_1, y_1), \ldots, f(x_k, y_k)}_{\text{demonstrations}}, f\left(\underbrace{x_{k+1}}_{\text{input}}, \underbrace{\quad}_{\text{answer}}\right)\right) \to \hat{y}_{k+1},$$

where $I$ is the task description, and $x_{k+1}$ is a new input query. It has been observed that the effectiveness of ICL during inference is sensitive to a variety of factors concerning both the instruction formatting and the demonstration organization. The former pertains to the realm of prompt engineering [17] and includes a wide range of possibilities (e.g. *Chain-of-Thought* prompting, role-playing strategies) that are beyond the scope of this work. For what concerns demonstration organization, three main aspects have been identified:

- **Demonstration Selection**: selecting a subset of examples from a given knowledge base. This can be done either for a given test instance or in order to select examples to annotate *before* test time. Both heuristic and LLM-based approaches have been proposed;

- **Demonstration Format**: effectively integrating and formatting each selected demonstration into a prompt formulated in natural language;
- **Demonstration Order**: rearranging the demonstrations in a good order. Several heuristic methods have been proposed, such as ordering the examples according to the similarity to the query in the embedding space.

### 2.1.1 Related Work

In this work we focus specifically on the Demonstration Selection aspect of ICL. It has been shown that choosing examples that are close to the query in the embedding space (either via cosine similarity or euclidean distance) can greatly improve the performance of ICL. This is due to the fact that closeness in the embedding space is associated to semantic similarity between the available demonstrations and the given query. In particular, in [16] the authors introduce and evaluate *KATE*, a KNN-based demonstration retriever, with GPT-3 in three different tasks, namely Sentiment Analysis, Table-to-text Generation and open-domain Question Answering. In this article, they use different sentence embeddings produced by both the original RoBERTa-large model [18] and differently fine-tuned versions of the same RoBERTa model. In order to retrieve similar examples, both euclidean distance and cosine similarity are considered. This approach has been shown to significantly outperform a random sampling baseline, and to achieve similar (or even better, in the case of Question Answering) performance to a small (3B parameters) fine-tuned T5 model on the same tasks.

In addition to the relevance to a given test instance, understood as closeness in the embedding space, semantic diversity between the retrieved examples has also been identified as a key factor for achieving good performance in

ICL. In [25] the authors identify a tension between the need for the demonstrations to be relevant to the test instance and the need for diversity between the examples. In the same article a reinforcement learning approach to demonstration selection is proposed, which aims to maximize both relevance and diversity. This approach, however, computes diversity by only taking into account the distribution of labels among the selected demonstrations.

In [28] it is shown that LLMs can benefit from exemplar sets that exhibit both complementarity and relevance to a given test query. Contrary to the previous study, the diversity between demonstrations does not concerns the labels only, but it is based on the same similarity metric that is used in order to assess the relevance. Specifically, they test a Maximum-Marginal-Relevance(MMR)-based retriever on three tasks (Letter Concatenation, Coin Flips and Grade-School Math). In particular, they use a greedy approach by iteratively selecting demonstrations in such a way to maximize the MMR:

$$\hat{q} = \arg\max_{q_j \in D/T} \left( \lambda \mathcal{S}(q, q_j) - (1 - \lambda) \max_{q_i \in T} \mathcal{S}(q_j, q_i) \right),$$

where $\mathcal{S}$ denotes a similarity function, $0 \leq \lambda \leq 1$ is a parameter which controls the trade-off between relevance and diversity, $D$ is the pool of available demonstrations and $T$ is the set of the currently selected demonstrations. It is observed that the best results are achieved for values $0.5 \leq \lambda \leq 0.6$, roughly balancing the impact of relevance and diversity.

Last, in [24] a graph-based approach is used in order to select effective demonstrations to label from a larger set of unlabeled data *before* test time. At test time, the retrieved demonstrations to be prompted to the LLM are still selected via KNN. This method also encourages diversity between the demonstrations, and has been proven to outperform random selection on several tasks.

## 2.2 Graph Theory

In this section we introduce several tools and concepts from graph theory upon which our approach to Demonstration Selection is grounded. Specifically, we make heavy use of the PageRank centrality measure and the Louvain method for community detection. We begin with the following basic definitions:

**Definition 2.2.1.** A *graph* $G$ is an ordered pair $G = (V, E)$, where $V$ is a finite set, whose elements are said *vertices* or *nodes*, and $E \subseteq V \times V$ is the set of the *edges* of the graph.

Edges can also be weighted according to some *weight function* $\omega \colon E \to \mathbb{R}$.

**Definition 2.2.2.** A graph $S = (V_S, E_S)$ is a *subgraph* of $G = (V, E)$ if $V_S \subseteq V$ and $E_S \subseteq E$.

In the following, we will label nodes in vertices with natural numbers $V = \{1, 2, \dots, n\}$, with $n = |V|$. Any graph with such labels can be uniquely identified by its *adjacency matrix*:

**Definition 2.2.3.** Let $G = (V, E)$ be a graph with $V = \{1, 2, \dots, n\}$. The *adjacency matrix* of $G$ is $A = (a_{ij}) \in \mathbb{R}^{n \times n}$, where

$$a_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

This definition naturally extends to weighted graphs by letting

$$a_{ij} = \begin{cases} \omega(i,j) & \text{if } (i,j) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

### 2.2.1 Centrality measures

In graph theory and network science, a common interesting property to study is which nodes can be deemed the most *important* in a given graph. Since there are a number of different characteristics that can be used to determine this importance, a number of *centrality measures* have been identified [19, 10, 4]. Formally, a centrality measure is any function of nodes that is invariant by graph automorphism:

**Definition 2.2.4.** Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs. Then $G_1$ and $G_2$ are *isomorphic* if and only if there exists a bijection $\phi\colon V_1 \to V_2$ such that for all $i, j \in V_1$,

$$(i, j) \in E_1 \iff (\phi(i), \phi(j)) \in E_2.$$

Such bijection $\phi$ is an *isomorphism* between $G_1$ and $G_2$. An isomorphism from a graph $G$ to itself $\phi\colon V \to V$ is said *automorphism*.

*Remark* 2.2.1. Two graphs $G_1, G_2$ are isomorphic if and only if there exists a permutation matrix $P$ such that their respective adjacency matrices $A_1, A_2$ are conjugated via $P$:

$$A_2 = PA_1P^T.$$

**Definition 2.2.5.** A *centrality measure* for a graph $G = (V, E)$ is a function $f_G\colon V \to \mathbb{R}$ such that for all graph automorphisms $\phi\colon V \to V$, and for all $i \in V$, $f_G(i) = f_G(\phi(i))$.

The definition of centrality measure is remarkably broad and allows to define measures that capture both local and global properties of graphs. For instance, one of the simplest centrality measures is the (out-)degree of nodes:

$$deg(i) = |\{(i, j) \mid (i, j) \in E\}|.$$

The class of centrality measures that is most relevant to this work is that of *spectral centrality measures*. This type of centrality measures are based on spectral properties of adjacency matrices. Often, they are based on the idea that the importance of a node correlates with the importance of neighboring nodes [22], and they account for long-range effects of nodes, rather than the effects on their immediate neighborhood. The most straightforward spectral measure is probably the *eigenvector centrality*. This measure assigns to each node $i$ a score $v_i$ that is proportional to the scores of nodes that are connected to it:

$$\lambda v_i = \sum_{j\,:\, j \to i} v_j = \sum_{j=1}^{n} A_{ji} v_j = \left(A^T \mathbf{v}\right)_i ,$$

where $A$ is the adjacency matrix of the graph. This means that the vector of scores $\mathbf{v}$ is in fact an eigenvector of $A^T$. For connected graphs, the Perron-Frobenius theorem [15] guarantees that choosing $\lambda$ as the largest eigenvalue and normalizing $\mathbf{v}$ makes this centrality measure well-defined:

**Definition 2.2.6.** A matrix $A \in \mathbb{R}^{n \times n}$ is *reducible* if there exists a permutation $P$ such that

$$A = P \begin{bmatrix} X & Y \\ 0 & Z \end{bmatrix} P^T,$$

where $X$ and $Z$ are both square. If a square matrix is not reducible then it is *irreducible*.

**Definition 2.2.7.** A graph $G = (V, E)$ is *connected* if for all $u, v \in V$, there exists a path from $u$ to $v$.

*Remark* 2.2.2. If $G$ is a connected graph, then its adjacency matrix is irreducible.

**Theorem 2.2.1** (Perron-Frobenius). *Let $A \in \mathbb{R}^{n \times n}$ be an irreducible and non-negative matrix. Then:*

1. *$A$ has an eigenvalue $\lambda = \max_{i=1,\dots,n} |\lambda_i|$, where $\lambda_1, \dots, \lambda_n$ are the eigenvalues of A;*

2. *there exists a positive eigenvector $\mathbf{v}$ associated to $\lambda$;*

3. *$\lambda$ has algebraic multiplicity 1;*

4. *if* $A\mathbf{x} = \lambda\mathbf{x}$*, then* $\mathbf{x}$ *is a multiple of* $\mathbf{v}$.

In this work, however, we deal for the most part with disconnected graphs. For this reason, we cannot use directly this measure when assessing the centrality of nodes. Another important spectral centrality measure that we can use in its place is the *PageRank* score [21]. This measure was defined by the founders of Google Larry Page and Sergey Brin. The main intuition that motivates it is very similar to the idea of mutual reinforcement behind the eigenvector centrality. In this case, the importance of a node is still measured in terms of the importance of its neighbors, but the contribution of each neighbor is divided by their respective (out-)degree. The behaviour of this metric on a directed graph is illustrated in Figure 2.2. In addition, the PageRank score is well-defined even for disconnected graphs, which allows it to be used in a wider range of situations.

**Definition 2.2.8.** Let $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ be the adjacency matrix of graph $G$. Then we define the *transition matrix* $N \in \mathbb{R}^{n\times n}$ of $G$ as

$$N = (n_{ij}) = \left(\frac{a_{ij}}{\sum_{k=1}^{n} a_{ik}}\right),$$

that is, the matrix obtained from the adjacency matrix by dividing each element by the sum of the elements in its row.

Formally, the vector of PageRank scores of a graph $G$ can be defined as the eigenvector centrality of the weighted graph that has as adjacency matrix a convex linear combination of its transition matrix and the constant matrix $\left(\frac{1}{n}\right)^{n\times n}$:

$$\Gamma = \alpha N + (1-\alpha)\left(\frac{1}{n}\right)^{n\times n},$$

where $0 < \alpha < 1$ is traditionally set to $\alpha = 0.85$. The matrix $\Gamma$ is called the *Google matrix* of graph $G$. By the Perron-Frobenius theorem, since this matrix is strictly positive, the vector of PageRank scores is well-defined for arbitrary graphs with nonnegative weights. Moreover, this measure is closely related

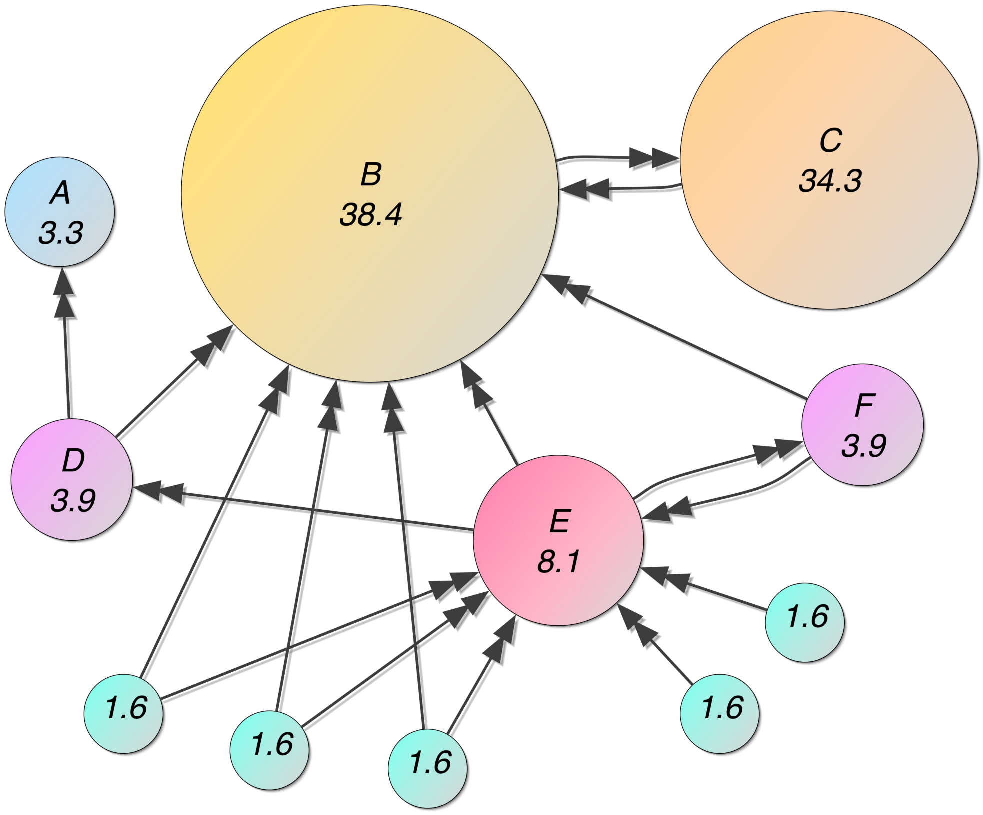

Figure 2.2: Graphical representation of unnormalized PageRank scores on a directed graph.

to the behaviour of random walk on graphs; in fact, it can be proved that it is the only stationary probability vector of the random walk on $G$ defined by the Google matrix.

### 2.2.2 Modularity

One of the most crucial problems in graph theory is to identify meaningful *communities* (or *clusters*) of nodes within graphs. For instance, if a graph is used to represent a net of friendships or co-authorships in research papers, communities can be formed by groups of friends that get on well or authors that are interested in the same specific topic. In general, divisions into communities of the vertices of a graph $G = (V, E)$ are subsets $C \subseteq \mathcal{P}(V)$, but they are often straight partitions of $V$. Usually, communities are groups of nodes that are densely connected to each other, and are scarcely connected to nodes belonging to other communities. A remarkably successful formalization of this concept is given in [20] and [8], via the definition of *modularity*. The modularity of a partition $\mathcal{C}$ of a graph is defined as

$$Q(\mathcal{C}) = \frac{1}{2m} \sum_{vw} \left[ A_{vw} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w),$$

where $m = \frac{1}{2} \sum_{vw} A_{vw}$ is the total number of edges in the graph, $A$ is as usual the adjacency matrix, $\delta(i, j)$ is $1$ if $i = j$ and $0$ otherwise, and $c_v, c_w \in \mathcal{C}$ are the communities that contain nodes $v$ and $w$, respectively. The term $\frac{k_v k_w}{2m}$ represents the probability of an edge existing between vertices $v$ and $w$ if connections are made at random but respecting vertex degrees. Hence, modularity can be interpreted as a measure of the ratio of intra-community edges to inter-community edges.

**Theorem 2.2.2.** *Let $G = (V, E)$ be an undirected, unweighted graph and $\mathcal{C} \subseteq \mathcal{P}(V)$ a partition of the nodes of $G$. Then, $-\frac{1}{2} \leq Q(\mathcal{C}) \leq 1$ holds.*

*Proof.* The theorem is proved in [6]. □

In particular, anti-community partitionings yield negative values for modularity, with the minimum being achieved for instance by bipartite graphs with the canonic clustering. If the number of intra-community edges is close to the expected value for a random graph with the same degree distribution, then $Q(\mathcal{C}) \approx 0$, and positive values indicate strong community structure.

### 2.2.3 Louvain method

It has been proved that the problem of finding the partition yielding the maximum value for modularity is strongly NP-complete [5]. One of the most popular and widely used heuristic methods for finding partitions with large modularity is the *Louvain method* [3]. Its popularity is due to the fact that it is extremely fast in terms of computational time, in addition to yielding clusters with good quality, modularity-wise. The algorithm is composed by two phases, repeated iteratively. In the first phase, it assigns each node to its own singleton community. Then, for each node $i$, it evaluates the gain of modularity that would be obtained by moving $i$ from its community to the ones of each of its neighbours. Last, if a positive gain is possible, $i$ is moved to the community that yields the largest modularity increase. The process is repeated for all nodes until no further improvement can be achieved. In the second phase, it builds a new graph whose nodes represent the communities indentified in the first phase. The new nodes are subsequently linked with edges having a weight given by the sum of the weights of the edges between nodes in the corresponding two communities (defaulting to 1 if the original network is unweighted). This leads to the creation of self-loops for each node having a weight equal to the sum of weights of intra-community edges in the corresponding community in the original graph. The two phases are iterated until no further gain of modularity is obtained. Two iterations of this process are illustrated in Figure 2.3.

*Remark* 2.2.3. In addition to returning a good partition of the whole graph into communities, the outlined process also yields a number of smaller, hierarchical subcommunities after each second phase.

*Remark* 2.2.4. The Louvain algorithm is sensitive to the order in which nodes are examinated and it does not provide a heuristic method to determine such order. Unless some criterium for ordering nodes is chosen beforehand, this process is not deterministic.

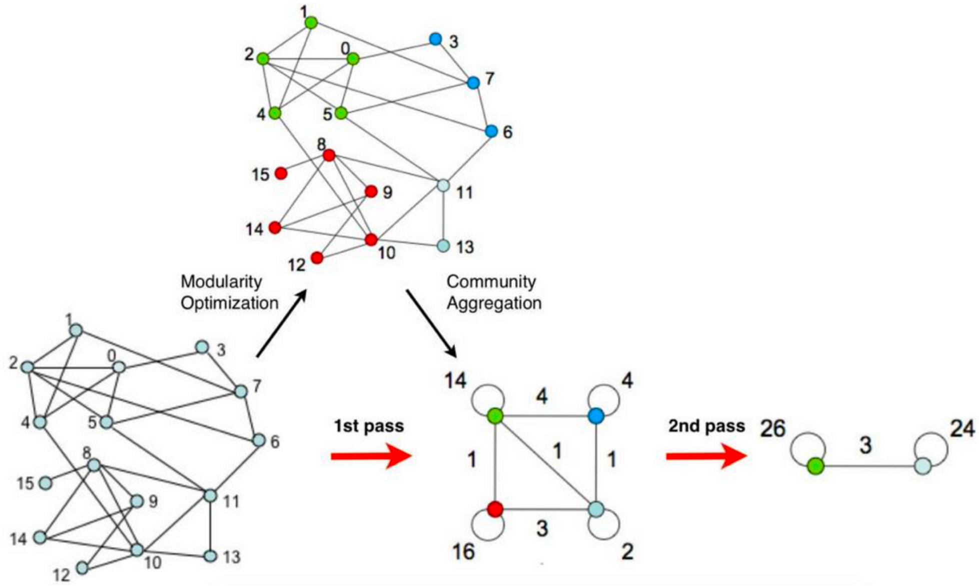

Figure 2.3: Graphical representation of the Louvain algorithm.

# Chapter 3

# Methodology

In this chapter we outline a general-purpose graph-based approach to demonstration retrieval for in-context learning. Guided by the observations from the sources described in Section 2.1 and by using the tools from Section 2.2 we introduce a method that retrieves demonstrations from an available knowledge base while pursuing the following goals:

**G1** The demonstrations must be relevant to any given input query;

**G2** In order to make the most use of the whole knowledge base, the demonstrations must be representative of abstract concepts represented in the knowledge base itself;

**G3** The demonstrations must be semantically diverse from each other in order to effectively cover a variety of concepts.

## 3.1 Demonstration Graph creation

The first step consists in the creation of the *Demonstration Graph* $G$. During this phase, for each demonstration $d_i$with features $\mathbf{x}_i$ in the knowledge base $\mathcal{K} = \{d_1, \dots, d_N\}$, a node with label $i$ is added to the Demonstration Graph. I Then, given a similarity measure $\mathcal{S}$ (or a distance $\mathcal{D}$) and a threshold $R$, for

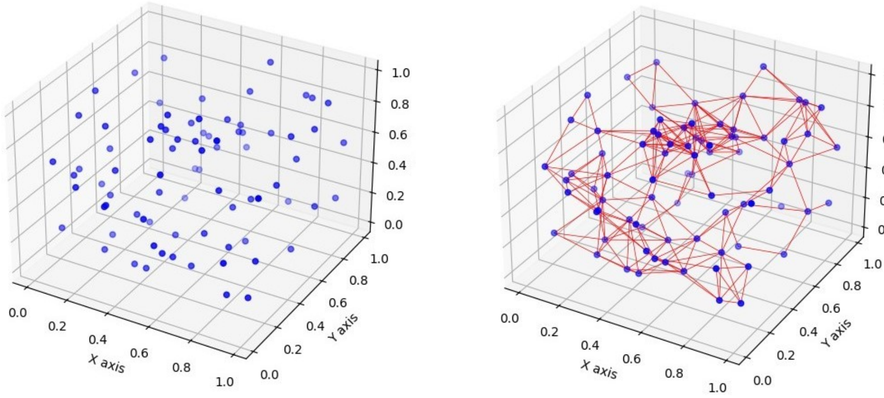

Figure 3.1: Demonstration Graph creation. In the left picture, blue points represent features of demonstrations from a knowledge base in a 3D space. In the right picture the same points are connected to each other if their euclidean distance is below a set threshold $R$.

each pair $1 \leq i, j \leq N$, an undirected edge is added between nodes $i$ and $j$ if and only if $\mathcal{S}(\mathbf{x}_i, \mathbf{x}_j) \geq R$ (respectively, $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_j) \leq R$).

*Remark* 3.1.1. The threshold parameter $R$ directly influences the edge density of the Demonstration Graph. In the following, we refer to it as the *resolution* of the graph. Large values of $R$ trivially correspond to highly dense Demonstration Graphs.

In the experiments from Chapter 4, we used both the Euclidean distance between sentence embeddings generated by the `Llama-3.1-8B-Instruct` transformer model and the Hamming distance between structured sequences of numbers as metrics for the creation of the Demonstration Graphs.

## 3.2 Query Subgraph Extraction

The second step is dependent on the query input. In this phase, given a test instance $q$ with features $\mathbf{x}_q$, a Demonstration Graph $G$ and a *radius* parameter $r$, we extract a subgraph $S$ of $G$ by removing from it each node $i$ such that $\mathcal{S}(\mathbf{x}_i, \mathbf{x}_q) \leq r$ (or $\mathcal{D}(\mathbf{x}_i, \mathbf{x}_q) \geq r$). This step is graphically illustrated in Figure 3.2 and Figure 3.3.

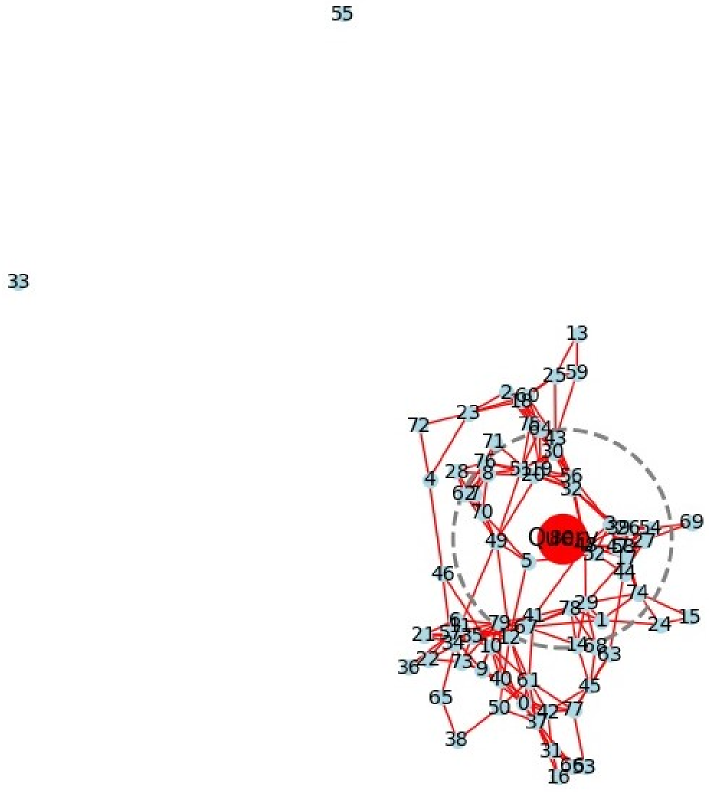

Figure 3.2: Flattened representation of the graph from Figure 3.1. The large red point represents the features of the query and the dashed circle represents the cut which yields the subgraph $S$.

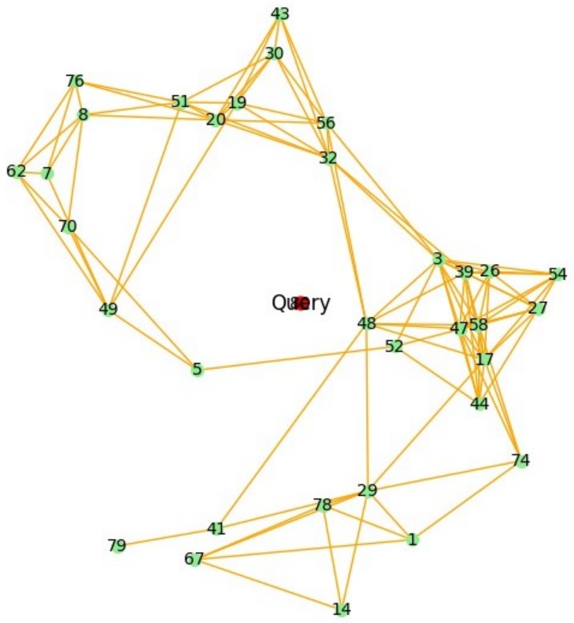

Figure 3.3: Extracted subgraph $S$ from the Demonstration Graph $G$ in Figure 3.2.

*Remark* 3.2.1. The first and second step are theoretically interchangeable. In fact, creating a large Demonstration Graph and then extracting a subgraph is equivalent to first selecting all nodes that are closer than $r$ to the query and then connecting them to each other according to the resolution parameter $R$. In the setting of our experiments, given the large number of required inferences, the first approach is more convenient. On the contrary, in a different scenario where the number of inferences is limited, the second approach might be more appropriate, since the former involves computing all the $\binom{N}{2}$ possible (dis)similarities and keeping a potentially much larger graph in memory.

*Remark* 3.2.2. Having an effect similar to the KNN approach, this step is responsible for achieving the goal **G1**. In particular, the lower the radius $r$, the more semantically relevant to the query the retrieved demonstrations will be.

*Remark* 3.2.3. Depending on the specific query instance, it is possible that the extracted subgraph has less nodes than the required number of demonstrations.

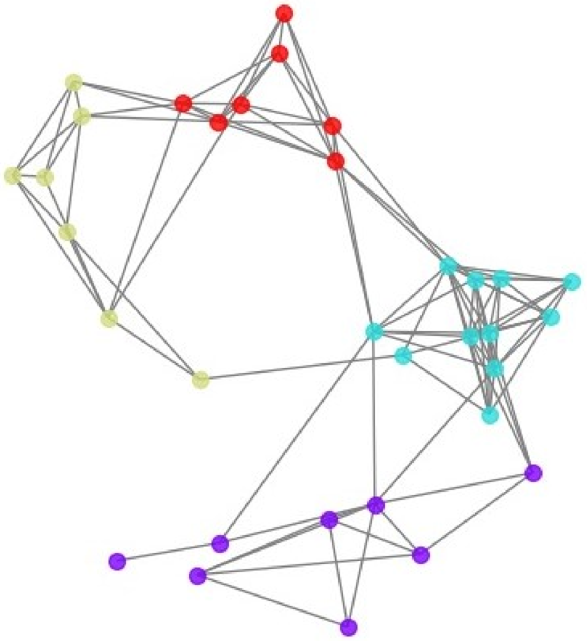

Figure 3.4: Partition of the graph from Figure 3.3 yielded by the Louvain algorithm.

In these cases, in the experiments we iteratively increase the radius $r$ by a relatively small amount (typically 10%) until a sufficient number of nodes is obtained.

*Remark* 3.2.4. If $G$ and $S$ are obtained by using a distance $\mathcal{D}$, then the resolution parameter $R$ should be set to less than double $r$. Otherwise, due to the triangular inequality, $S$ would result in a clique, rendering subsequent steps useless.

## 3.3 Louvain Partitioning and Demonstration Selection

The last step consists in creating partitions in the subgraph $S$ and selecting the most *important* nodes from each partition. First we apply the Louvain method on $S$ in order to detect cohesive partitions.

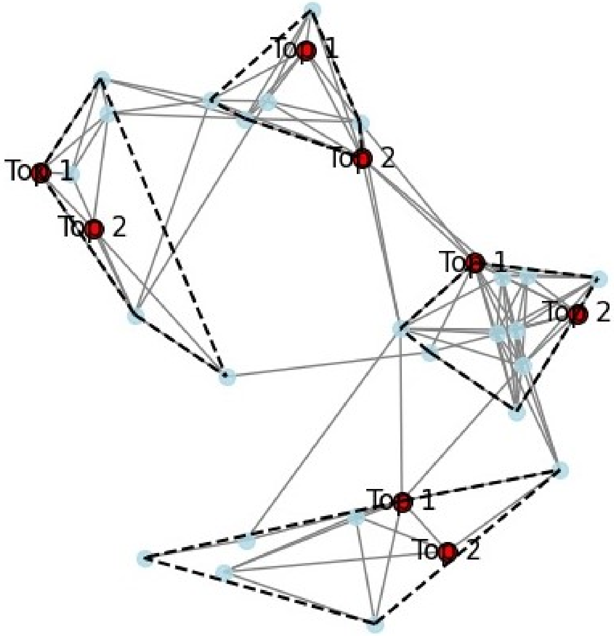

Figure 3.5: Selection of the top 2 nodes according to the PageRank metric computed in each community detected in Figure 3.4.

*Remark* 3.3.1. The application of the Louvain algorithm on the extracted subgraph allows the satisfaction of the objective **G3**. In fact, by choosing nodes from disjunct and dense communities, we obtain demonstrations that are related to separate abstract concepts.

Last, from each community we select the $k$ nodes that have the highest PageRank score computed in the respective partition.

*Remark* 3.3.2. The Louvain method does not allow to decide beforehand how many communities are detected. In the experiments, we select the top $k$ nodes from the largest detected communities, where $k$ is the smallest integer that allows to obtain a sufficient number of demonstrations.

*Remark* 3.3.3. Following the discussion from Section 2.2.1, in this context where edges between nodes represent similarity between demonstrations, nodes with a high PageRank score represent examples that are similar to many examples that are in turn similar to many demonstrations. Hence, these examples make for good representatives of abstract concepts encoded in their respective partition, working towards the satisfaction of goal **G2**.

## 3.4 Label-Balanced Variant

We also test a variant of the described method specifically for single-label and multi-label classification. In this case, we aim to further increase the diversity of demonstrations by providing in the prompt an approximatively equal number of examples for each possible label. By following the outlined methodology, it is possible that all of the retrieved demonstrations share the same label, especially for low values of the radius $r$ and for highly unbalanced datasets. This has the potential to set major drawbacks in the performance of ICL: preliminary experiments show that not having access to any example for a given label can make discarding it remarkably difficult for the LLM, unless the description of the label itself in the template is extremely accurate and effective. This variant simply consists in creating a different Demonstration Graph for each label in the dataset by following the same process as before. Then, for each graph we retrieve the most relevant demonstrations and include them in the prompt.

*Remark* 3.4.1. For multi-label classification, this can lead to including multiple copies of some demonstrations in the prompt. If the number $L$ of labels is very small (ideally $2$ or $3$, depending on how large the knowledge base is and on the actual distribution of the labels included in it), another possibility that allows to avoid this phenomenon would be to create a different Demonstration Graph for each possible combination of labels. Of course, the exponential growth of the combinations renders this approach unviable for larger numbers of labels.

Figures 3.6 and 3.7 illustrate an example of how the two methods differ on a test instance from our first experiment.

**Query**:

Text: finding corroborated Article 20 Regulation 6591999 governs rights interested parties

Answer:

**Retrieved Examples**:

Text: Secondly reasons set paragraph 65 present judgment General Court required adjudicate discounts provided 2006 schedule

Answer: **prem**

Text: result aid cannot considered separately method financing see effect judgment 14 April 2005 AEM AEM Torino C-12803 C-12903 EUC2005224 paragraph 45

Answer: **prem**

Text: apparent Court's caselaw cited paragraphs 89 90 recovery unlawful aid different purpose Directive 2014104

Answer: **prem**

Text: Second regards merits plea Article 44 Regulation 178581 merely provides Articles 92 93 94 Treaty apply production trade sugar save otherwise provided Regulation

Answer: **prem**

Text: Accordingly General Court err law determining reference system

Answer: **prem**

**LLM output: conc**

**ground truth: prem**

Figure 3.6: Example of prompt with demonstrations retrieved via the original graph-based approach. The examples are not able to provide sufficient informations to the LLM in order to exclude the "conc" label.

**Query**:
Text: finding corroborated Article 20 Regulation 6591999 governs rights interested parties
Answer:

**Retrieved Examples**:
Text: follows first ground appeal must rejected part inadmissible part unfounded
Answer: **conc**
Text: result aid cannot considered separately method financing see effect judgment 14 April 2005 AEM AEM Torino C-12803 C-12903 EUC2005224 paragraph 45
Answer: **prem**
Text: circumstances third ground appeal must rejected unfounded
Answer: **conc**
Text: follows foregoing second plea relied upon Telefónica support appeal must also dismissed
Answer: **conc**
Text: Secondly reasons set paragraph 65 present judgment General Court required adjudicate discounts provided 2006 schedule
Answer: **prem**

**LLM output: prem**
**ground truth: prem**

Figure 3.7: Example of prompt with demonstrations retrieved by using a separate graph for each label. Contrary to the example from Figure 3.6, the demonstrations with the “conc” label guide the LLM to output the correct answer.

# Chapter 4

# Experiments

We apply the described methodology to five tasks tasks on three different datasets. In all tasks, we join a predefined template with 5, 10, 15, 30 and 50 retrieved examples and provide the resulting prompt as input to the transformer model `Llama-3.1-8B-Instruct`. We perform experiments by setting the resolution $R$ and radius $r$ depending on the specific dataset: we test resolutions that yield Demonstration Graphs where each node is connected on average to 0.5% (*"Sparse Graphs"*, in the Tables), 1% (*"Default $R$"*) and 2% (*"Dense Graphs"*) of other nodes, and subgraph radii such that, applied to demonstrations in the knowledge base, would yield subgraphs of average size $\sqrt{N \cdot 50}$ (*"Default r"*, in the Tables) and $\frac{\sqrt{N \cdot 50}}{2}$ (*"Small r"*), where $N$ is the number of examples in the knowledge base. This last choice is motivated by the fact that in this way, when choosing 50 demonstrations, the two steps of subgraph extraction and selection of the most *important* nodes in the detected communities yield on average two roughly equal reductions of the size of the available dataset. Moreover, as an attempt to give even more importance to similarity in the graph-based approach, we test using half the default radius (*"Smallest r"*). This always yields empty graphs, until the iterative process of incrementing $r$ yields enough nodes. In all experiments we compare our approach to two baselines with random and KNN-based demonstration selection. For the classification tasks we include an additional Zero-Shot baseline.

The first dataset is Demosthenes [13, 23], a corpus on which we tackle three sentence classification tasks from the field of argument mining in the legal domain. The second dataset [12] includes annotated privacy policies of online platforms, on which we classify clauses as *sufficienty informative* or *insufficiently informative* according to Articles 13 and 14 of the GDPR. Last, we perform experiments on the capability of the LLM of understanding the legality of moves from examples in the *Chef's Hat* boardgame [1, 2]. In this case we use a dataset that has been artificially created by letting agents play random moves for 100 matches.

In the appendix we show all the prompt templates that have been used in the experiments.

## 4.1 Argument Mining

### 4.1.1 Dataset

The Demosthenes corpus encompasses 40 decisions on fiscal State aids by the Court of Justice of the European Union (CJEU). The dataset is characterized by a focus on argument mining, with 4 related classification tasks being addressed. Sentences from the *Findings of the court* section of each document have been manually annotated by experts in the legal domain. The choice of this specific section is motivated by the fact that it has been identified as the main source of interacting inferences, which ultimately lead to conclusions on the parties' claims. The documents have been pre-processed via the removal of stop-words and punctuation and the sentence segmentation has been performed based on periods, semicolons and newlines. The tasks introduced in the article that presents the corpus are:

- Argument Detection;
- Argument Classification;

- Type Classification;
- Scheme Classification.

We tackle here the last three tasks, which assume that argumentative components have been previously correctly identified. In the original article, the authors evaluated several traditional machine learning techniques via 5-fold cross-validation, with manually created splits at the document level in order to balance their composition. In our experiments, we use one of these folds as a test set, reserving all the other elements to the knowledge base. For all the tasks, in order to measure the dissimilarity of demonstrations, we use the euclidean distance between sentence embeddings produced by the same model used for inference.

### 4.1.2 Argument Classification

This first single-label classification task consists in determining whether a sentence that has been identified as argumentative is a *premise* (*prem*) or the *conclusion* (*conc*) of the argument it belongs to. It is important to point out that, in general, an argumentative sentence can be both a premise of an argument and a conclusion of another argument at the same time. In these cases, sentences have been marked as *premises*. The dataset for this task is highly unbalanced, including 2535 argumentative sentences, of which 2375 are marked as premises and 160 marked as conclusions. In the test set there are 345 premises and 22 conclusions.

Table 4.1 shows the F1 scores achieved by the different approaches and Table 4.2 shows the same metric achieved by the respective label-balanced variants. The most dramatic improvement in performances comes by switching each approach to its respective label-balanced version. In fact, on average, the base methods achieve a macro F1 score lower than $10$, with exception of the Random-based demonstration retrieval, which achieves the highest F1 score for both labels on average. However, even the Random-based approach fails in achieving performances comparable to the Zero-Shot approach. This is most probably due to the high unbalance of labels: the KNN-based and graph-based approaches are more likely to retrieve only *premise*-labeled examples to each of the test instances with the same label, resulting in a severe degradation of performance.

All the label-balanced variants yield a dramatic improvement, with the strongest performance being obtained by the KNN-based retrieval approach. In almost all cases the performances tend to increase with respect to the number of provided demonstrations. falling behind the competitive KNN baseline, the label-balanced variant of the graph-based approach surpasses the zero-shot and random baselines in all configurations, with the best improvements being achieved with extracted subgraphs of small size. These considerations indicate that the proposed method can be beneficial, even if the impact of similarity has been underestimated.

### 4.1.3 Type Classification

The second task we tackle is Type Classification. This is a multi-label classification problem in which premises have to be classified as *legal* (*L*) and/or *factual* (*F*). Legal premises are sentences that support a conclusion by providing legal elements such as legal rules, precedents, interpretations of applicable rules and principles; while premises labeled as factual contain descriptions of specific events and existing situations. Of the 2375 premises in the corpus, 906 have been marked as *legal*, and 1576 as *factual*, with only 107 possessing both labels.

The performance of the considered base and label-balanced approaches are shown in Table 4.3 and Table 4.4, respectively. Interestingly, in this case the base approaches seem to slightly outperform the label-balanced variants. In general, however, the number of demonstrations and the methods to retrieve them do not seem to affect significantly the performances. Remarkably, in

| Method | # Dem. | Premise | Conclusion | Macro |
|---|---|---|---|---|
| **Zero-Shot** | | **96.88** | **21.43** | **59.16** |
| **Random** | 5 | 17.46 | 12.36 | 14.91 |
| | 10 | 11.99 | 11.99 | 11.99 |
| | 15 | 8.86 | 11.80 | 10.33 |
| | 30 | 6.18 | 11.64 | 8.91 |
| | 50 | 12.50 | 12.02 | 12.26 |
| | Avg. | 11.40 | 11.96 | 11.68 |
| **KNN** | 5 | 8.09 | 6.06 | 7.07 |
| | 10 | 7.65 | 8.15 | 7.90 |
| | 15 | 8.22 | 9.21 | 8.72 |
| | 30 | 9.78 | 9.29 | 9.54 |
| | 50 | 14.89 | 10.61 | 12.75 |
| | Avg. | 9.73 | 8.66 | 9.20 |

| Method | # Dem. | Premise | Conclusion | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Default r** | 5 | 8.72 | 8.72 | 8.72 |
| | 10 | 6.13 | 10.13 | 8.13 |
| | 15 | 6.69 | 10.67 | 8.68 |
| | 30 | 8.26 | 10.24 | 9.25 |
| | 50 | 7.76 | 10.72 | 9.24 |
| | Avg. | 7.52 | 10.10 | 8.80 |
| **Louvain; Dense Graph; Default r** | 5 | 8.77 | 9.76 | 9.26 |
| | 10 | 3.94 | 10.03 | 6.99 |
| | 15 | 4.46 | 8.53 | 6.50 |
| | 30 | 5.56 | 9.09 | 7.32 |
| | 50 | 8.29 | 10.75 | 9.52 |
| | Avg. | 6.20 | 9.63 | 7.92 |
| **Louvain; Sparse Graph; Default r** | 5 | 8.15 | 7.65 | 7.90 |
| | 10 | 5.04 | 10.08 | 7.56 |
| | 15 | 6.67 | 10.16 | 8.41 |
| | 30 | 10.87 | 10.38 | 10.63 |
| | 50 | 15.75 | 9.07 | 12.41 |
| | Avg. | 9.30 | 9.47 | 9.38 |

| Method | # Dem. | Premise | Conclusion | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Small r** | 5 | 10.19 | 7.20 | 8.69 |
| | 10 | 5.54 | 8.58 | 7.06 |
| | 15 | 6.63 | 9.14 | 7.88 |
| | 30 | 8.74 | 9.24 | 8.99 |
| | 50 | 7.73 | 10.22 | 8.97 |
| | Avg. | 7.77 | 8.88 | 8.32 |
| **Louvain; Dense Graph; Small r** | 5 | 6.63 | 9.14 | 7.88 |
| | 10 | 5.01 | 9.07 | 7.04 |
| | 15 | 5.56 | 9.09 | 7.32 |
| | 30 | 6.69 | 10.67 | 8.68 |
| | 50 | 6.65 | 9.65 | 8.15 |
| | Avg. | 6.11 | 9.52 | 7.81 |
| **Louvain; Sparse Graph; Small r** | 5 | 7.18 | 9.68 | 8.43 |
| | 10 | 6.61 | 8.63 | 7.63 |
| | 15 | 10.30 | 9.32 | 9.81 |
| | 30 | 10.84 | 9.86 | 10.35 |
| | 50 | 13.33 | 9.47 | 11.40 |
| | Avg. | 9.65 | 9.39 | 9.52 |

| Method | # Dem. | Premise | Conclusion | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Smallest r** | 5 | 8.70 | 8.20 | 8.45 |
| | 10 | 7.61 | 7.10 | 7.36 |
| | 15 | 7.67 | 8.67 | 8.17 |
| | 30 | 5.00 | 8.56 | 6.78 |
| | 50 | 9.32 | 10.30 | 9.81 |
| | Avg. | 7.66 | 8.57 | 8.11 |
| **Louvain; Dense Graph; Smallest r** | 5 | 9.16 | 7.16 | 8.16 |
| | 10 | 5.51 | 7.55 | 6.53 |
| | 15 | 7.65 | 8.15 | 7.90 |
| | 30 | 4.47 | 9.04 | 6.76 |
| | 50 | 7.18 | 9.68 | 8.43 |
| | Avg. | 6.79 | 8.32 | 7.56 |
| **Louvain; Sparse Graph; Smallest r** | 5 | 7.65 | 8.15 | 7.90 |
| | 10 | 7.67 | 8.67 | 8.17 |
| | 15 | 10.81 | 9.34 | 10.08 |
| | 30 | 9.81 | 9.81 | 9.81 |
| | 50 | 9.86 | 10.84 | 10.35 |
| | Avg. | 9.16 | 9.36 | 9.26 |

Table 4.1: F1 scores for Zero-Shot, Random-based, KNN-based and graph-based approaches on the Argument Classification task without balancing the labels in the prompt.

| Method | # Dem. | Premise | Conclusion | Macro |
|---|---|---|---|---|
| **Random** | 5 | 76.21 | 24.00 | 50.10 |
| | 10 | 85.38 | 33.33 | 59.36 |
| | 15 | 86.14 | 34.38 | 60.26 |
| | 30 | 86.93 | 34.43 | 60.68 |
| | 50 | 88.92 | 37.84 | 63.38 |
| | Avg. | 84.72 | 32.80 | 58.76 |
| **KNN** | 5 | 87.50 | 29.09 | 58.30 |
| | 10 | 93.42 | 46.91 | 70.16 |
| | 15 | 92.76 | 44.71 | 68.73 |
| | 30 | 92.74 | 45.98 | 69.36 |
| | 50 | **94.03** | **51.85** | **72.94** |
| | Avg. | 92.09 | 43.71 | 67.90 |

| Method | # Dem. | Premise | Conclusion | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Default r** | 5 | 83.44 | 23.08 | 53.26 |
| | 10 | 88.36 | 31.78 | 60.07 |
| | 15 | 91.67 | 37.21 | 64.44 |
| | 30 | 91.25 | 40.43 | 65.84 |
| | 50 | 91.78 | 40.45 | 66.12 |
| | Avg. | 89.30 | 34.59 | 61.95 |
| **Louvain; Dense Graph; Default r** | 5 | 82.35 | 24.46 | 53.41 |
| | 10 | 88.68 | 33.64 | 61.16 |
| | 15 | 92.76 | 44.71 | 68.73 |
| | 30 | 89.03 | 34.29 | 61.66 |
| | 50 | 91.28 | 39.13 | 65.20 |
| | Avg. | 88.82 | 35.25 | 62.03 |
| **Louvain; Sparse Graph; Default r** | 5 | 81.34 | 20.14 | 50.74 |
| | 10 | 86.82 | 26.79 | 56.80 |
| | 15 | 91.47 | 38.20 | 64.84 |
| | 30 | 91.56 | 42.55 | 67.06 |
| | 50 | 92.64 | 41.46 | 67.05 |
| | Avg. | 88.77 | 33.83 | 61.30 |

| Method | # Dem. | Premise | Conclusion | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Small r** | 5 | 81.62 | 22.70 | 52.16 |
| | 10 | 87.76 | 26.67 | 57.21 |
| | 15 | 91.61 | 40.00 | 65.81 |
| | 30 | 91.30 | 37.78 | 64.54 |
| | 50 | 92.09 | 42.79 | 67.39 |
| | Avg. | .8888 | 33.99 | 61.42 |
| **Louvain; Dense Graph; Small r** | 5 | 84.87 | 26.98 | 55.93 |
| | 10 | 88.82 | 28.28 | 58.55 |
| | 15 | 92.09 | 42.70 | 67.39 |
| | 30 | 90.88 | 40.82 | 65.86 |
| | 50 | 91.05 | 41.24 | 66.14 |
| | Avg. | 89.54 | 36.00 | 62.77 |
| **Louvain; Sparse Graph; Small r** | 5 | 82.00 | 19.40 | 50.70 |
| | 10 | 88.00 | 31.19 | 59.60 |
| | 15 | 92.09 | 42.70 | 67.39 |
| | 30 | 90.85 | 42.00 | 66.43 |
| | 50 | 92.57 | 45.45 | 69.01 |
| | Avg. | 89.10 | 36.15 | 62.63 |

| Method | # Dem. | Premise | Conclusion | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Smallest r** | 5 | 83.44 | 27.83 | 57.21 |
| | 10 | 90.42 | 37.11 | 63.77 |
| | 15 | 92.74 | 45.98 | 69.36 |
| | 30 | 92.19 | 46.81 | 69.50 |
| | 50 | 93.70 | 50.60 | 72.15 |
| | Avg. | 90.50 | 41.67 | 66.40 |
| **Louvain; Dense Graph; Smallest r** | 5 | 86.41 | 27.59 | 57.00 |
| | 10 | 89.45 | 32.32 | 60.89 |
| | 15 | 93.42 | 46.91 | 70.16 |
| | 30 | 91.73 | 43.01 | 67.37 |
| | 50 | 92.86 | 48.89 | 70.87 |
| | Avg. | 90.77 | 39.74 | 65.26 |
| **Louvain; Sparse Graph; Smallest r** | 5 | 86.45 | 26.32 | 56.38 |
| | 10 | 89.91 | 36.00 | 62.95 |
| | 15 | 93.72 | 49.38 | 71.55 |
| | 30 | 91.88 | 44.68 | 68.28 |
| | 50 | 93.70 | 50.60 | 72.15 |
| | Avg. | 91.13 | 41.40 | 66.26 |

Table 4.2: F1 scores for Random-based, KNN-based and graph-based approaches on the Argument Classification task by balancing the labels in the prompt.

most cases a larger number of examples seem to slightly increase the F1 score for the *factual* label, while having the opposite effect on both the macro F1 score and the F1 score for the *legal* label. The best macro F1 scores are obtained in both cases by different variants of the graph-based approach with a small amount of demonstrations ($\sim$ **71** and $\sim$ **70**, respectively), while the KNN-based methods achieve the strongest F1 on the *factual* label when retrieving many examples. The best average macro F1 scores are achieved by the graph-based approach with the default value for the subgraph radius $r$, even if by tiny margins, suggesting that for this task the increase in semantic diversity between the demonstrations is slightly more beneficial and impactful than similarity. It is worth observing that for this task the Zero-Shot approach obtains a relatively strong performance, indicating that in this specific case the demonstrations are not particularly effective in guiding the model towards correct classifications.

### 4.1.4 Scheme Classification

The last task we tackle on the Demosthenes corpus is Scheme Classification. This is again a multi-label classification task, in which the goal is to establish whether a legal premise is used in an inference that follows one of the following argumentation schemes:

- Rule (or established rule) scheme;
- Precedent scheme;
- Authoritative scheme;
- Classification scheme;
- Interpretative scheme.

The *Rule* scheme characterizes legal premises that explicitly cite an EU norm as part of the relevant legislative framework. Legal premises belonging to

| Method | # Dem. | Factual | Legal | Macro |
|---|---|---|---|---|
| **Zero-Shot** |  | 80.21 | 59.24 | 69.73 |
| **Random** | 5 | 80.78 | 57.21 | 68.99 |
|  | 10 | 80.07 | 60.29 | 70.18 |
|  | 15 | 79.86 | 61.20 | 70.53 |
|  | 30 | 80.14 | 58.17 | 69.15 |
|  | 50 | 80.07 | 57.02 | 68.54 |
|  | Avg. | 80.18 | 58.78 | 69.48 |
| **KNN** | 5 | 80.21 | 59.39 | 69.80 |
|  | 10 | 79.72 | 59.50 | 69.61 |
|  | 15 | 80.00 | 58.35 | 69.18 |
|  | 30 | 80.84 | 57.83 | 69.34 |
|  | 50 | **81.51** | 57.74 | 69.62 |
|  | Avg. | 80.46 | 58.56 | 69.51 |

| Method | # Dem. | Factual | Legal | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Default r** | 5 | 78.86 | 57.51 | 68.18 |
|  | 10 | 80.28 | 59.11 | 69.70 |
|  | 15 | 80.21 | 59.31 | 69.76 |
|  | 30 | 79.79 | 58.85 | 69.32 |
|  | 50 | 81.21 | 58.93 | 70.07 |
|  | Avg. | 80.07 | 58.74 | 69.41 |
| **Louvain; Dense Graph; Default r** | 5 | 79.14 | 59.65 | 69.40 |
|  | 10 | 79.72 | 61.19 | 70.46 |
|  | 15 | 80.00 | 60.62 | 70.31 |
|  | 30 | 79.37 | 60.61 | 69.99 |
|  | 50 | 80.57 | 59.95 | 70.26 |
|  | Avg. | 79.76 | 60.40 | 70.08 |
| **Louvain; Sparse Graph; Default r** | 5 | 79.86 | 59.80 | 69.83 |
|  | 10 | 79.50 | 57.99 | 68.74 |
|  | 15 | 79.86 | 56.04 | 67.95 |
|  | 30 | 79.93 | 58.00 | 68.97 |
|  | 50 | 80.49 | 57.59 | 69.04 |
|  | Avg. | 79.93 | 57.88 | 68.91 |

| Method | # Dem. | Factual | Legal | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Small r** | 5 | 79.57 | 60.05 | 69.81 |
|  | 10 | 79.08 | 60.67 | 69.87 |
|  | 15 | 81.28 | 58.94 | 70.11 |
|  | 30 | 80.63 | 59.11 | 69.87 |
|  | 50 | 80.43 | 58.47 | 69.45 |
|  | Avg. | 80.20 | 59.45 | 69.82 |
| **Louvain; Dense Graph; Small r** | 5 | 79.86 | 60.05 | 69.95 |
|  | 10 | 80.36 | **61.35** | **70.85** |
|  | 15 | 80.00 | 58.27 | 69.13 |
|  | 30 | 80.57 | 58.82 | 69.69 |
|  | 50 | 80.28 | 57.68 | 68.98 |
|  | Avg. | 80.21 | 59.23 | 69.72 |
| **Louvain; Sparse Graph; Small r** | 5 | 80.14 | 59.66 | 69.90 |
|  | 10 | 80.00 | 57.84 | 68.92 |
|  | 15 | 80.71 | 58.39 | 69.55 |
|  | 30 | 80.14 | 58.60 | 69.37 |
|  | 50 | 80.99 | 58.47 | 69.73 |
|  | Avg. | 80.40 | 58.59 | 69.49 |

| Method | # Dem. | Factual | Legal | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Smallest r** | 5 | 80.21 | 59.75 | 69.98 |
|  | 10 | 79.50 | 58.44 | 68.97 |
|  | 15 | 80.35 | 60.50 | 70.43 |
|  | 30 | 79.86 | 59.13 | 69.50 |
|  | 50 | 80.84 | 58.26 | 69.55 |
|  | Avg. | 80.15 | 59.22 | 69.69 |
| **Louvain; Dense Graph; Smallest r** | 5 | 80.78 | 59.50 | 70.14 |
|  | 10 | 79.93 | 57.79 | 68.86 |
|  | 15 | 78.57 | 59.54 | 69.06 |
|  | 30 | 80.35 | 58.45 | 69.40 |
|  | 50 | 80.71 | 56.42 | 68.57 |
|  | Avg. | 80.07 | 58.34 | 69.21 |
| **Louvain; Sparse Graph; Smallest r** | 5 | 79.86 | 59.20 | 69.53 |
|  | 10 | 79.21 | 59.30 | 69.25 |
|  | 15 | 79.35 | 58.42 | 68.88 |
|  | 30 | 80.63 | 56.59 | 68.61 |
|  | 50 | 81.36 | 57.60 | 69.48 |
|  | Avg. | 80.08 | 58.22 | 69.15 |

Table 4.3: F1 scores for Zero-Shot, Random-based, KNN-based and graph-based approaches on the Type Classification task without balancing the labels in the prompt. The underlined configurations are the ones that perform better than both the Random and the KNN-based retrieval approaches in terms of average Macro F1 score.

| Method | # Dem. | Factual | Legal | Macro |
|---|---|---|---|---|
| **Random** | 5 | 79.93 | 58.88 | 69.40 |
| | 10 | 79.79 | 56.88 | 68.33 |
| | 15 | 80.35 | 58.45 | 69.40 |
| | 30 | 80.21 | 56.22 | 68.22 |
| | 50 | 80.49 | 56.05 | 68.27 |
| | Avg. | 80.15 | 57.30 | 68.72 |
| **KNN** | 5 | 80.71 | 59.75 | 70.23 |
| | 10 | 80.78 | 58.31 | 69.54 |
| | 15 | 80.70 | 56.67 | 68.68 |
| | 30 | 80.56 | 55.73 | 68.15 |
| | 50 | **81.91** | 56.33 | 69.12 |
| | Avg. | 80.93 | 57.36 | 69.14 |

| Method | # Dem. | Factual | Legal | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Default r** | 5 | 79.86 | **60.47** | 70.16 |
| | 10 | 80.70 | 58.59 | 69.64 |
| | 15 | 80.35 | 58.25 | 69.30 |
| | 30 | 80.57 | 56.76 | 68.66 |
| | 50 | 80.28 | 56.47 | 68.37 |
| | Avg. | 80.35 | 58.11 | 69.23 |
| **Louvain; Dense Graph; Default r** | 5 | 80.71 | 59.41 | 70.06 |
| | 10 | 79.08 | 58.37 | 68.73 |
| | 15 | 80.07 | 58.12 | 69.10 |
| | 30 | 81.13 | 55.95 | 68.54 |
| | 50 | 80.56 | 55.72 | 68.14 |
| | Avg. | 80.31 | 57.51 | 68.91 |
| **Louvain; Sparse Graph; Default r** | 5 | 80.56 | 58.11 | 69.34 |
| | 10 | 80.50 | 57.14 | 68.82 |
| | 15 | 80.42 | 56.80 | 68.61 |
| | 30 | 80.42 | 56.43 | 68.43 |
| | 50 | 80.42 | 55.60 | 68.01 |
| | Avg. | 80.46 | 56.82 | 68.64 |

| Method | # Dem. | Factual | Legal | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Small r** | 5 | 81.35 | 59.20 | **70.28** |
| | 10 | 80.14 | 57.76 | 68.95 |
| | 15 | 80.70 | 55.71 | 68.21 |
| | 30 | 80.49 | 55.90 | 68.19 |
| | 50 | 80.57 | 55.79 | 68.18 |
| | Avg. | 80.65 | 56.87 | 68.76 |
| **Louvain; Dense Graph; Small r** | 5 | 80.35 | 59.85 | 70.10 |
| | 10 | 80.56 | 57.77 | 69.16 |
| | 15 | 80.70 | 57.35 | 69.02 |
| | 30 | 80.35 | 55.51 | 67.93 |
| | 50 | 80.21 | 55.72 | 67.97 |
| | Avg. | 80.43 | 57.24 | 68.84 |
| **Louvain; Sparse Graph; Small r** | 5 | 81.13 | 58.44 | 69.78 |
| | 10 | 80.35 | 57.89 | 69.12 |
| | 15 | 80.28 | 57.14 | 68.71 |
| | 30 | 80.42 | 56.00 | 68.21 |
| | 50 | 80.78 | 56.21 | 68.49 |
| | Avg. | 80.59 | 57.14 | 68.86 |

| Method | # Dem. | Factual | Legal | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Smallest r** | 5 | 79.93 | 58.50 | 69.21 |
| | 10 | 80.28 | 56.80 | 68.54 |
| | 15 | 80.92 | 56.54 | 68.73 |
| | 30 | 80.42 | 57.40 | 68.91 |
| | 50 | 81.48 | 55.13 | 68.30 |
| | Avg. | 80.61 | 56.87 | 68.74 |
| **Louvain; Dense Graph; Smallest r** | 5 | 79.72 | 59.90 | 69.81 |
| | 10 | 80.14 | 57.97 | 69.06 |
| | 15 | 80.28 | 56.74 | 68.51 |
| | 30 | 80.42 | 56.64 | 68.53 |
| | 50 | 80.63 | 56.17 | 68.40 |
| | Avg. | 80.24 | 57.48 | 68.86 |
| **Louvain; Sparse Graph; Smallest r** | 5 | 80.57 | 56.87 | 68.72 |
| | 10 | 80.85 | 58.03 | 69.44 |
| | 15 | 81.77 | 57.21 | 69.49 |
| | 30 | 80.28 | 56.70 | 68.49 |
| | 50 | 80.84 | 56.28 | 68.56 |
| | Avg. | 80.86 | 57.02 | 68.94 |

Table 4.4: F1 scores for Random-based, KNN-based and graph-based approaches on the Type Classification task by balancing the labels in the prompt. The underlined configuration performs better than both the Random and the KNN-based retrieval approaches in terms of average Macro F1 score.

the Premise scheme (*Prem*) refer to past decisions of the CJEU. Authoritative (*Aut*) legal premises include references to indications by an authority, not necessarily legally binding, such as opinions of the Advocate General. Sentences annotated under the Classification (*Class*) scheme consist of definitions of legal concepts. Last, the Interpretative (*Itpr*) scheme ascribes a meaning relevant to the decision to a legal source via various kinds of interpretative reasoning, such as literal, teleological or psychological interpretation of the legal source. In addition to these argumentative schemes, in [13] the authors mention the Principle scheme, which applies when a general legal principle is applicable to a case and may determine its outcome. However, this last scheme has not been considered due to it not being sufficiently represented in the dataset. Of the mentioned schemes, only the *Authoritative*, *Precedent* and *Rule* are to be considered reliably annotated, due to the strong agreement between the two annotators of the dataset, while the others are mentioned to be potentially noisy. Of the legal premises, 53 are marked as belonging to the *Authoritative* scheme, 503 belong to the *Precedent* scheme and 322 to the *Rule* scheme. For *unreliable* schemes, there are 56 sentences belonging to the *Classification* scheme and 296 belonging to the *Interpretative* scheme.

The F1 scores for each class are displayed in Tables from 4.5 to 4.12 for both the base and the label-balanced approaches. The base methods' results are comparable to Zero-Shot classification (which achieves a reliable macro F1 score of $\sim \mathbf{70}$), while balancing the labels in the retrieved demonstrations yields a major improvement. Generally, increasing the number of retrieved demonstrations seem to enhance the performances, especially concerning the *reliable* macro F1. For label-balanced graph-based approaches, denser Demonstration Graphs work better on average for the *default* and *small* values of $r$, while the opposite is true for very small extracted subgraphs. The best *reliable* macro F1 score ($\sim \mathbf{85}$) is achieved by the label-balanced version of KNN with 15 demonstrations. On average however, the results of this last approach are matched by the label-balanced graph-based method with a *small*

| Method | # Dem. | Aut | Class | Itpr | Prec | Rule | Macro | Macro (reliable) |
|---|---|---|---|---|---|---|---|---|
| **Zero-Shot** | | 66.67 | 0.00 | 22.86 | 84.44 | 59.52 | 46.70 | 70.21 |
| **Random** | 5 | 57.14 | 0.00 | 33.90 | 78.61 | 62.59 | 46.45 | 66.11 |
| | 10 | 57.14 | 11.76 | 31.75 | 77.71 | 61.15 | 47.90 | 65.33 |
| | 15 | 60.00 | 0.00 | 28.13 | 75.98 | 60.00 | 44.82 | 65.33 |
| | 30 | 66.67 | 0.00 | 24.14 | 79.56 | 64.15 | 46.90 | 70.13 |
| | 50 | 85.71 | 0.00 | 21.05 | 77.84 | 60.00 | 48.92 | 74.52 |
| | Avg. | 65.33 | 2.35 | 27.79 | 77.94 | 61.58 | 47.00 | 68.28 |
| **KNN** | 5 | 47.06 | 40.00 | 40.74 | 84.09 | **68.46** | 56.07 | 66.54 |
| | 10 | 57.14 | **62.50** | 38.46 | 82.95 | 65.38 | 61.29 | 68.49 |
| | 15 | 66.67 | 58.82 | 39.22 | 82.22 | 67.11 | 62.81 | 72.00 |
| | 30 | 80.00 | 47.62 | 32.14 | 82.61 | 64.90 | 61.45 | 75.84 |
| | 50 | **100.00** | 50.00 | 42.31 | 82.87 | 64.10 | **67.86** | **82.33** |
| | Avg. | 70.17 | 51.79 | 38.57 | 82.95 | 65.99 | 61.90 | 73.04 |

Table 4.5: F1 scores for Zero-Shot, Random-based and KNN-based approaches on the Scheme Classification task without balancing the labels in the prompt.

subgraph radius $r$ and a *dense* Demonstration Graph. Similarly to what has been observed for the first task, the graph-based approaches greatly overcome the Zero-Shot and Random baselines, but in most cases they fall slightly behind the KNN-based approach, pointing towards a slight underestimation of the impact of similarity in favor of the improved semantic diversity.

## 4.2 Privacy Policy Compliance

### 4.2.1 Dataset

The second dataset we experiment with has been presented in [12]. This corpus has a strong focus on assessing the comprehensiveness of information provided by data controllers to data subjects in privacy policies. The corpus contains 30 privacy policies of online companies and in the original article the authors performed a manual train-validation-test split at the document level, with rate 60%-20%-20%. In order to mantain coherence with the original study, in this experiment we use the same training set (with sentences from

| Method | # Dem. | Aut | Class | Itpr | Prec | Rule | Macro | Macro (reliable) |
|---|---|---|---|---|---|---|---|---|
| **Louvain;<br>Default R;<br>Default r** | 5 | 80.00 | 27.27 | 11.76 | 77.58 | 60.76 | 51.47 | 72.78 |
| | 10 | 57.14 | 47.06 | 19.35 | 80.70 | 62.11 | 53.27 | 66.65 |
| | 15 | 80.00 | 40.00 | 29.51 | 77.46 | 61.35 | 57.66 | 72.94 |
| | 30 | 80.00 | 42.11 | 26.67 | 78.61 | 61.35 | 57.75 | 73.32 |
| | 50 | 57.14 | 38.10 | 30.30 | 76.74 | 61.73 | 52.80 | 65.21 |
| | Avg. | 70.86 | 38.91 | 23.52 | 78.22 | 61.46 | 54.59 | 70.18 |
| **Louvain;<br>Dense Graph;<br>Default r** | 5 | 72.73 | 30.00 | 27.12 | 77.11 | 62.89 | 53.97 | 70.91 |
| | 10 | 66.67 | 28.57 | 34.92 | 80.92 | 64.43 | 55.10 | 70.67 |
| | 15 | 66.67 | 47.06 | 35.82 | 77.65 | 59.63 | 57.36 | 67.98 |
| | 30 | 66.67 | 40.00 | 26.09 | 79.77 | 62.34 | 54.97 | 69.59 |
| | 50 | 66.67 | 44.44 | 19.72 | 75.58 | 65.81 | 54.44 | 69.35 |
| | Avg. | 67.88 | 38.01 | 28.73 | 78.21 | 63.02 | 55.17 | 69.70 |
| **Louvain;<br>Sparse Graph;<br>Default r** | 5 | 72.73 | 47.06 | 31.25 | 75.74 | 62.42 | 57.84 | 70.30 |
| | 10 | 72.73 | 57.14 | 36.84 | 84.66 | 59.74 | 62.22 | 72.38 |
| | 15 | 57.14 | 57.14 | 24.24 | 83.04 | 59.34 | 56.18 | 66.51 |
| | 30 | 80.00 | 53.33 | 26.47 | 83.15 | 64.90 | 61.57 | 76.02 |
| | 50 | 80.00 | 47.06 | 35.09 | 84.32 | 63.23 | 61.94 | 75.85 |
| | Avg. | 72.52 | 52.35 | 30.78 | 82.18 | 61.93 | 59.95 | 72.21 |

Table 4.6: F1 scores for the graph-based approach with Default $r$ on the Scheme Classification task without balancing the labels in the prompt.

| Method | # Dem. | Aut | Class | Itpr | Prec | Rule | Macro | Macro (reliable) |
|---|---|---|---|---|---|---|---|---|
| **Louvain;<br>Default R;<br>Small r** | 5 | 66.67 | 42.11 | 25.00 | 75.31 | 63.29 | 54.47 | 68.42 |
| | 10 | 80.00 | 40.00 | 33.90 | 82.29 | 63.29 | 59.90 | 75.19 |
| | 15 | 88.89 | 31.58 | 31.03 | 79.10 | 60.87 | 58.29 | 76.28 |
| | 30 | 75.00 | 50.00 | 31.75 | 80.43 | 64.47 | 60.33 | 73.30 |
| | 50 | 72.73 | 53.33 | 30.99 | 82.22 | 66.23 | 61.10 | 73.72 |
| | Avg. | 76.66 | 43.40 | 30.53 | 79.87 | 63.63 | 58.82 | 73.38 |
| **Louvain;<br>Dense Graph;<br>Small r** | 5 | 47.06 | 26.09 | 25.00 | 82.14 | 63.23 | 48.70 | 64.14 |
| | 10 | 54.55 | 42.11 | 31.88 | 78.61 | 60.26 | 53.48 | 64.47 |
| | 15 | 42.86 | 42.11 | 34.78 | 79.31 | 60.65 | 51.94 | 60.94 |
| | 30 | 72.73 | 38.10 | 33.33 | 81.97 | 63.58 | 57.94 | 72.76 |
| | 50 | 72.73 | 47.06 | 39.29 | 84.78 | 64.52 | 61.67 | 74.01 |
| | Avg. | 57.99 | 39.09 | 32.86 | 81.36 | 62.45 | 54.75 | 67.26 |
| **Louvain;<br>Sparse Graph;<br>Small r** | 5 | 44.44 | 31.58 | 30.19 | 80.92 | 61.04 | 49.64 | 62.14 |
| | 10 | 57.14 | 44.44 | 23.33 | 82.49 | 65.31 | 54.54 | 68.31 |
| | 15 | 72.73 | 55.56 | 36.92 | 82.02 | 64.86 | 62.42 | 73.20 |
| | 30 | 80.00 | 50.00 | **50.00** | 83.42 | 63.51 | 65.39 | 75.65 |
| | 50 | 80.00 | 58.82 | 37.74 | 85.41 | 64.90 | 65.37 | 76.77 |
| | Avg. | 66.86 | 48.08 | 35.64 | 82.85 | 63.92 | 59.47 | 71.21 |

Table 4.7: F1 scores for the graph-based approach with Small $r$ on the Scheme Classification task without balancing the labels in the prompt. The underlined configuration performs better than both the Random and the KNN-based retrieval approaches in terms of average Macro (reliable) F1 score.

| Method | # Dem. | Aut | Class | Itpr | Prec | Rule | Macro | Macro (reliable) |
|---|---|---|---|---|---|---|---|---|
| **Louvain; Default R; Smallest r** | 5 | 53.33 | 32.00 | 35.71 | 81.61 | 64.00 | 53.33 | 66.31 |
| | 10 | 40.00 | 30.00 | 28.07 | 82.95 | 61.84 | 48.57 | 61.60 |
| | 15 | 50.00 | 52.63 | 32.35 | 81.61 | 62.34 | 55.79 | 64.65 |
| | 30 | 72.73 | 55.56 | 33.33 | 84.27 | 66.67 | 62.51 | 74.55 |
| | 50 | 85.71 | 50.00 | 42.62 | 80.43 | 65.36 | 64.83 | 77.17 |
| | Avg. | 60.35 | 44.04 | 34.42 | 82.17 | 64.04 | 57.01 | 68.86 |
| **Louvain; Dense Graph; Smallest r** | 5 | 53.33 | 42.11 | 37.04 | 81.40 | 64.52 | 55.68 | 66.41 |
| | 10 | 61.54 | 47.62 | 29.63 | 79.31 | 64.05 | 56.43 | 68.30 |
| | 15 | 61.54 | 43.48 | 33.90 | 84.75 | 64.86 | 57.71 | 70.38 |
| | 30 | 72.73 | 55.56 | 26.42 | 82.95 | 64.52 | 60.43 | 73.40 |
| | 50 | 66.67 | 47.06 | 33.33 | 84.78 | 65.77 | 59.52 | 72.41 |
| | Avg. | 63.16 | 47.17 | 32.06 | 82.64 | 64.74 | 57.95 | 70.18 |
| **Louvain; Sparse Graph; Smallest r** | 5 | 57.14 | 50.00 | 45.28 | **86.86** | 64.56 | 60.77 | 69.52 |
| | 10 | 57.14 | 50.00 | 37.04 | 82.76 | 66.67 | 58.72 | 68.86 |
| | 15 | 72.73 | 58.82 | 31.03 | 84.62 | 66.67 | 62.77 | 74.67 |
| | 30 | 88.89 | 50.00 | 43.64 | 84.32 | 65.36 | 66.44 | 79.52 |
| | 50 | 80.00 | 43.48 | 33.96 | 84.78 | 65.33 | 61.51 | 76.71 |
| | Avg. | 71.81 | 50.46 | 38.19 | 84.67 | 65.72 | 62.04 | 73.86 |

Table 4.8: F1 scores for the graph-based approach with Smallest $r$ on the Scheme Classification task without balancing the labels in the prompt. The underlined configurations performs better than both the Random and the KNN-based retrieval approaches in terms of average Macro (reliable) F1 score.

| Method | # Dem. | Aut | Class | Itpr | Prec | Rule | Macro | Macro (reliable) |
|---|---|---|---|---|---|---|---|---|
| **Random** | 5 | 57.14 | 0.00 | 31.03 | 81.40 | 63.09 | 46.53 | 67.21 |
| | 10 | 60.00 | 12.90 | 20.34 | 79.04 | 61.25 | 46.71 | 66.76 |
| | 15 | 85.71 | 16.22 | 26.67 | 77.91 | 64.00 | 54.10 | 75.87 |
| | 30 | 88.89 | 23.81 | 13.33 | 74.29 | 65.75 | 53.21 | 76.31 |
| | 50 | **100.00** | 29.41 | 12.50 | 74.29 | 64.52 | 56.14 | 79.60 |
| | Avg. | 78.35 | 16.47 | 20.77 | 77.39 | 63.72 | 51.34 | 73.15 |
| **KNN** | 5 | **100.00** | 22.22 | 37.04 | 82.95 | 66.67 | 61.78 | 83.21 |
| | 10 | 80.00 | **34.48** | 31.03 | 83.15 | 66.67 | 59.07 | 76.60 |
| | 15 | **100.00** | 31.25 | **42.86** | 85.88 | 68.49 | **65.70** | **84.79** |
| | 30 | 88.89 | **34.48** | 27.59 | 85.56 | 69.50 | 61.20 | 81.32 |
| | 50 | **100.00** | 29.41 | 28.57 | 81.61 | **70.92** | 62.10 | 84.18 |
| | Avg. | 93.78 | 30.37 | 33.42 | 83.83 | 68.45 | 61.97 | 82.02 |

Table 4.9: F1 scores for Random-based, KNN-based and graph-based approaches on the Scheme Classification task by balancing the labels in the prompt.

| Method | # Dem. | Aut | Class | Itpr | Prec | Rule | Macro | Macro (reliable) |
|---|---|---|---|---|---|---|---|---|
| **Louvain;<br>Default R;<br>Default r** | 5 | **100.00** | 20.69 | 33.96 | 83.24 | 63.16 | 60.21 | 82.13 |
| | 10 | 85.71 | 25.64 | 26.87 | 82.68 | 66.67 | 57.51 | 78.35 |
| | 15 | 57.14 | 21.62 | 20.59 | 82.02 | 62.75 | 48.82 | 67.30 |
| | 30 | **100.00** | 27.03 | 28.57 | 83.70 | 64.94 | 60.85 | 82.88 |
| | 50 | **100.00** | 31.25 | 18.92 | 81.11 | 66.23 | 59.50 | 82.45 |
| | Avg. | 88.57 | 25.25 | 25.78 | 82.55 | 64.75 | 57.38 | 78.62 |
| **Louvain;<br>Dense Graph;<br>Default r** | 5 | 80.00 | 30.77 | 20.69 | 80.00 | 62.11 | 54.71 | 74.04 |
| | 10 | **100.00** | 30.30 | 25.00 | 84.39 | 63.16 | 60.57 | 82.52 |
| | 15 | 80.00 | 30.30 | 21.33 | 85.56 | 66.20 | 56.68 | 77.25 |
| | 30 | 88.89 | **34.48** | 25.32 | 85.71 | 68.46 | 60.57 | 81.02 |
| | 50 | **100.00** | 24.39 | 15.79 | 77.27 | 66.21 | 56.73 | 81.16 |
| | Avg. | 89.78 | 30.05 | 21.63 | 82.59 | 65.23 | 57.85 | 79.20 |
| **Louvain;<br>Sparse Graph;<br>Default r** | 5 | 66.67 | 26.67 | 25.00 | 83.98 | 60.69 | 52.60 | 70.44 |
| | 10 | **100.00** | 25.64 | 22.54 | 84.27 | 68.61 | 60.21 | 84.29 |
| | 15 | 75.00 | 27.78 | 18.18 | 80.00 | 66.67 | 53.53 | 73.89 |
| | 30 | **100.00** | 25.00 | 17.65 | 84.66 | 69.06 | 59.27 | 84.57 |
| | 50 | 88.89 | 28.57 | 24.32 | 84.62 | 66.67 | 58.61 | 80.06 |
| | Avg. | 86.11 | 26.73 | 21.54 | 83.51 | 66.34 | 56.84 | 78.65 |

Table 4.10: F1 scores for the graph-based approach with Default $r$ on the Scheme Classification task by balancing the labels in the prompt.

| Method | # Dem. | Aut | Class | Itpr | Prec | Rule | Macro | Macro (reliable) |
|---|---|---|---|---|---|---|---|---|
| **Louvain;<br>Default R;<br>Small r** | 5 | 88.89 | 15.38 | 28.57 | 82.22 | 63.69 | 55.75 | 78.27 |
| | 10 | 75.00 | 27.59 | 26.23 | 86.03 | 62.34 | 55.44 | 74.46 |
| | 15 | 80.00 | 28.57 | 20.34 | 84.57 | 63.64 | 55.42 | 76.07 |
| | 30 | 80.00 | 27.78 | 19.35 | 84.92 | 65.75 | 55.56 | 76.89 |
| | 50 | 80.00 | 29.41 | 21.54 | 81.14 | 64.43 | 55.30 | 75.19 |
| | Avg. | 80.78 | 25.75 | 23.21 | 83.78 | 63.97 | 55.49 | 76.18 |
| **Louvain;<br>Dense Graph;<br>Small r** | 5 | **100.00** | 21.43 | 29.63 | 84.62 | 61.15 | 59.36 | 81.92 |
| | 10 | 88.89 | 33.33 | 25.00 | 84.09 | 63.01 | 58.87 | 78.66 |
| | 15 | **100.00** | 33.33 | 28.17 | 85.08 | 65.73 | 62.46 | 83.61 |
| | 30 | **100.00** | 30.30 | 22.54 | 84.57 | 66.67 | 60.82 | 83.75 |
| | 50 | **100.00** | 23.26 | 21.54 | 82.87 | 64.71 | 58.47 | 82.53 |
| | Avg. | 97.78 | 28.33 | 25.38 | 84.25 | 64.25 | 60.00 | 82.09 |
| **Louvain;<br>Sparse Graph;<br>Small r** | 5 | 66.67 | 22.22 | 26.23 | 85.71 | 64.86 | 53.14 | 72.42 |
| | 10 | 75.00 | 32.26 | 25.40 | 83.98 | 65.75 | 56.48 | 74.91 |
| | 15 | 75.00 | 26.32 | 24.62 | **87.64** | 67.12 | 56.14 | 76.59 |
| | 30 | 88.89 | 27.03 | 18.75 | 83.80 | 69.93 | 57.68 | 80.87 |
| | 50 | 88.89 | 27.03 | 25.40 | 84.09 | 64.43 | 57.97 | 79.14 |
| | Avg. | 78.89 | 26.97 | 24.08 | 85.04 | 66.42 | 56.28 | 76.79 |

Table 4.11: F1 scores for the graph-based approach with Small $r$ on the Scheme Classification task by balancing the labels in the prompt. The underlined configuration performs better than both the Random and the KNN-based retrieval approaches in terms of average Macro F1 score.

| Method | # Dem. | Aut | Class | Itpr | Prec | Rule | Macro | Macro (reliable) |
|---|---|---|---|---|---|---|---|---|
| **Louvain;<br>Default R;<br>Smallest r** | 5 | **100.00** | 25.81 | 40.00 | 86.52 | 62.42 | 62.95 | 82.98 |
| | 10 | 80.00 | 27.59 | 36.67 | 82.95 | 64.90 | 58.42 | 75.95 |
| | 15 | 80.00 | 30.30 | 30.51 | 85.23 | 66.67 | 58.54 | 80.00 |
| | 30 | 75.00 | 30.30. | 30.99 | 86.36 | 69.93 | 58.52 | 77.10 |
| | 50 | **100.00** | 27.03 | 22.95 | 80.23 | 69.86 | 60.01 | 83.36 |
| | Avg. | 87.00 | 28.21 | 32.22 | 84.26 | 66.76 | 59.69 | 79.88 |
| **Louvain;<br>Dense Graph;<br>Smallest r** | 5 | 88.89 | 32.26 | 35.29 | 84.92 | 64.10 | 61.09 | 79.30 |
| | 10 | 80.00 | 27.78 | 21.82 | 80.00 | 68.46 | 55.61 | 76.15 |
| | 15 | 85.71 | 31.25 | 27.59 | 82.29 | 67.11 | 58.79 | 78.37 |
| | 30 | 85.71 | 30.30. | 26.67 | 84.44 | 68.06 | 59.04 | 79.40 |
| | 50 | **100.00** | 30.30 | 28.57 | 82.49 | 70.00 | 62.27 | 84.16 |
| | Avg. | 88.06 | 30.38 | 27.99 | 82.83 | 67.55 | 59.36 | 79.48 |
| **Louvain;<br>Sparse Graph;<br>Smallest r** | 5 | 80.00 | 25.00 | 36.00 | 84.15 | 64.05 | 57.84 | 76.07 |
| | 10 | **100.00** | 31.25 | 28.07 | 85.39 | 68.46 | 62.63 | 84.62 |
| | 15 | 88.89 | 27.03 | 31.58 | 84.75 | 67.57 | 59.96 | 80.40 |
| | 30 | 57.14 | 30.30 | 29.51 | 80.45 | 67.57 | 52.99 | 68.39 |
| | 50 | **100.00** | 26.32 | 36.07 | 82.68 | 66.67 | 62.35 | 83.12 |
| | Avg. | 85.21 | 27.98 | 32.25 | 83.48 | 66.86 | 59.15 | 78.52 |

Table 4.12: F1 scores for the graph-based approach with Smallest $r$ on the Scheme Classification task by balancing the labels in the prompt.

18 documents) as the source of clauses for our knowledge base and we perform the evaluation of the various methods with the same test set (including 6 documents). The relevant clauses included in the dataset have been manually annotated as *sufficiently informative* (Level 1 in Tables) or *insufficiently informative* (Level 2 in Tables) according to the dispositions from Articles 13 and 14 of the GDPR. The task we tackle here is therefore a binary, single-label classification task. The dataset is remarkably unbalanced with respect to the labels: of the 579 relevant clauses in the dataset, 438 have been marked as *insufficiently informative*, with the remaining 141 marked as *sufficiently informative*. Contrary to Demosthenes, clauses included in this dataset have not undergone pre-processing (such as stop-words removal) except for text segmentation into sentences.

### 4.2.2 Experimental Results

Table 4.13 and Table 4.14 summarize the results achieved by the LLM with the different methods and configurations for demonstration retrieval. In this case, providing demonstrations proves to be greatly beneficial to ICL performance, with all methods and configurations surpassing the Zero-Shot baseline (macro F1 $\sim \mathbf{39}$) by ample margins. Moreover, the macro F1 scores vastly improve with larger numbers of provided examples and forcing an equal number of both labels in the retrieved demonstrations. In the non-balanced case, the random baseline is the best performing approach, on average. However, the random approach does not benefit as much from larger numbers of demonstrations, with the best single performance ($\sim \mathbf{65}$ macro F1 score) being achieved by the graph-based approach with 50 examples and a *Dense* Demonstration Graph. In the balanced case, the KNN-based approach with 50 demonstrations achieves the single strongest performance ($\sim \mathbf{74}$ F1 score), while being surpassed on average by the graph-based methods with *Default* $R$ and small subgraph sizes. This is mainly due to the high variance of the performance of KNN. In fact, this approach seem to be much less effective than the graph-based one when a low number of demonstration is provided, with a difference in macro F1 score of up to $\sim \mathbf{12}$ with 5 demonstrations. This phenomenon is likely to be due to the increased diversity provided by the graph-based methods even with a small number of examples: with few demonstrations the KNN approach fails in covering a sufficient variety of abstract concepts, while by increasing the number of demonstration (and, as a consequence, the average distance between the examples and the query) it gains a better capability of capturing more diverse and informative concepts.

## 4.3 Legal Moves Generation

The last experiment focuses on the task of understanding the legality of the moves of the Chef's Hat [1] card game via In-Context Learning. In particular, we do not provide an explanation of the rules of the game, and test whether and to what extent the LLM is able to recognize which moves are legal in a given game state having access only to a number of examples with pairs of

| Method | # Dem. | Level 1 | Level 2 | Macro |
|---|---|---|---|---|
| **Zero-Shot** | | 35.53 | 42.94 | 39.24 |
| **Random** | 5 | 44.29 | 56.67 | 50.48 |
| | 10 | 45.93 | 60.54 | 53.23 |
| | 15 | 40.00 | 56.22 | 48.11 |
| | 30 | 43.41 | 61.78 | 52.60 |
| | 50 | 45.05 | 70.81 | 57.93 |
| | Avg. | 43.74 | 61.20 | 52.47 |
| **KNN** | 5 | 39.74 | 46.15 | 42.94 |
| | 10 | 45.12 | 42.31 | 43.71 |
| | 15 | 42.50 | 42.50 | 42.50 |
| | 30 | 48.98 | 56.65 | 52.81 |
| | 50 | 48.48 | 63.83 | 56.16 |
| | Avg. | 44.96 | 50.29 | 47.62 |

| Method | # Dem. | Level 1 | Level 2 | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Default r** | 5 | 40.24 | 37.18 | 38.71 |
| | 10 | 44.58 | 40.26 | 42.42 |
| | 15 | 46.45 | 49.79 | 48.07 |
| | 30 | 43.36 | 54.24 | 48.80 |
| | 50 | 53.57 | 75.00 | 64.29 |
| | Avg. | 45.64 | 51.29 | 48.46 |
| **Louvain; Dense Graph; Default r** | 5 | 41.25 | 41.50 | 41.38 |
| | 10 | 45.86 | 47.85 | 46.86 |
| | 15 | 43.04 | 44.44 | 43.74 |
| | 30 | 44.44 | 49.10 | 46.77 |
| | 50 | **54.87** | **75.36** | **65.11** |
| | Avg. | 45.89 | 51.65 | 48.77 |
| **Louvain; Sparse Graph; Default r** | 5 | 43.59 | 46.34 | 44.97 |
| | 10 | 40.99 | 40.25 | 40.62 |
| | 15 | 45.28 | 45.96 | 45.62 |
| | 30 | 42.76 | 52.57 | 47.67 |
| | 50 | 53.10 | 74.40 | 63.75 |
| | Avg. | 45.14 | 51.90 | 48.53 |

| Method | # Dem. | Level 1 | Level 2 | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Small r** | 5 | 43.04 | 44.44 | 43.74 |
| | 10 | 44.31 | 39.22 | 41.76 |
| | 15 | 45.57 | 46.91 | 46.24 |
| | 30 | 48.00 | 54.12 | 51.06 |
| | 50 | 45.38 | 67.66 | 56.52 |
| | Avg. | 45.26 | 50.47 | 47.86 |
| **Louvain; Dense Graph; Small r** | 5 | 44.30 | 45.68 | 44.99 |
| | 10 | 45.00 | 45.00 | 45.00 |
| | 15 | 44.87 | 47.56 | 46.22 |
| | 30 | 46.90 | 56.00 | 51.45 |
| | 50 | 50.88 | 72.82 | 61.85 |
| | Avg. | 46.39 | 53.41 | 49.90 |
| **Louvain; Sparse Graph; Small r** | 5 | 44.44 | 43.04 | 43.74 |
| | 10 | 40.99 | 40.25 | 40.62 |
| | 15 | 44.59 | 46.63 | 45.61 |
| | 30 | 49.66 | 58.29 | 53.97 |
| | 50 | 51.24 | 70.35 | 60.80 |
| | Avg. | 46.18 | 51.71 | 48.95 |

| Method | # Dem. | Level 1 | Level 2 | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Smallest r** | 5 | 43.21 | 41.77 | 42.49 |
| | 10 | 45.12 | 42.31 | 43.71 |
| | 15 | 44.72 | 44.03 | 44.37 |
| | 30 | 48.95 | 58.76 | 53.85 |
| | 50 | 48.74 | 69.65 | 59.20 |
| | Avg. | 46.15 | 51.30 | 48.72 |
| **Louvain; Dense Graph; Smallest r** | 5 | 42.42 | 38.71 | 40.57 |
| | 10 | 43.68 | 32.88 | 38.28 |
| | 15 | 43.37 | 38.96 | 41.17 |
| | 30 | 44.44 | 54.55 | 49.49 |
| | 50 | 49.18 | 68.69 | 58.93 |
| | Avg. | 44.62 | 46.76 | 45.69 |
| **Louvain; Sparse Graph; Smallest r** | 5 | 42.24 | 41.51 | 41.87 |
| | 10 | 44.44 | 43.04 | 43.74 |
| | 15 | 43.04 | 44.44 | 43.74 |
| | 30 | 44.74 | 50.00 | 47.37 |
| | 50 | 50.42 | 70.65 | 60.53 |
| | Avg. | 44.98 | 49.93 | 47.45 |

Table 4.13: F1 scores for Zero-Shot, Random-based, KNN-based and graph-based approaches on the Privacy Policy dataset without balancing the labels in the prompt.

| Method | # Dem. | Level 1 | Level 2 | Macro |
|---|---|---|---|---|
| **Random** | 5 | 43.93 | 59.89 | 51.92 |
| | 10 | 47.14 | 58.10 | 52.62 |
| | 15 | 51.52 | 65.96 | 58.74 |
| | 30 | 51.13 | 65.24 | 58.18 |
| | 50 | 55.86 | 76.56 | 66.21 |
| | Avg. | 49.92 | 65.15 | 57.53 |
| **KNN** | 5 | 41.96 | 53.11 | 47.53 |
| | 10 | 48.92 | 60.77 | 54.85 |
| | 15 | 51.56 | 67.71 | 59.64 |
| | 30 | 54.55 | 76.19 | 65.37 |
| | 50 | **62.22** | **85.22** | **73.72** |
| | Avg. | 51.84 | 68.60 | 60.22 |

| Method | # Dem. | Level 1 | Level 2 | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Default r** | 5 | 52.71 | 68.06 | 60.39 |
| | 10 | 52.17 | 63.74 | 57.96 |
| | 15 | 50.37 | 63.78 | 57.08 |
| | 30 | 52.99 | 72.91 | 62.95 |
| | 50 | 50.00 | 68.37 | 59.18 |
| | Avg. | 51.65 | 67.37 | 59.51 |
| **Louvain; Dense Graph; Default r** | 5 | 48.53 | 61.96 | 55.24 |
| | 10 | 52.31 | 67.36 | 59.83 |
| | 15 | 46.27 | 61.29 | 53.78 |
| | 30 | 52.31 | 67.37 | 59.84 |
| | 50 | 50.82 | 69.70 | 60.26 |
| | Avg. | 50.05 | 65.54 | 57.79 |
| **Louvain; Sparse Graph; Default r** | 5 | 50.00 | 63.04 | 56.52 |
| | 10 | 49.25 | 63.44 | 56.35 |
| | 15 | 50.00 | 66.67 | 58.33 |
| | 30 | 57.63 | 75.25 | 66.44 |
| | 50 | 50.88 | 72.82 | 61.85 |
| | Avg. | 51.55 | 68.24 | 59.90 |

| Method | # Dem. | Level 1 | Level 2 | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Small r** | 5 | 51.09 | 63.39 | 57.24 |
| | 10 | 48.92 | 60.77 | 54.85 |
| | 15 | 53.85 | 68.42 | 61.13 |
| | 30 | 56.00 | 71.79 | 63.90 |
| | 50 | 57.66 | 77.51 | 67.58 |
| | Avg. | 53.50 | 68.38 | 60.94 |
| **Louvain; Dense Graph; Small r** | 5 | 50.38 | 65.61 | 58.00 |
| | 10 | 51.80 | 62.98 | 57.39 |
| | 15 | 52.48 | 62.57 | 57.53 |
| | 30 | 55.12 | 70.47 | 62.79 |
| | 50 | 52.73 | 75.24 | 63.98 |
| | Avg. | 52.50 | 67.37 | 59.94 |
| **Louvain; Sparse Graph; Small r** | 5 | 50.38 | 65.61 | 58.00 |
| | 10 | 46.04 | 58.56 | 52.30 |
| | 15 | 49.64 | 62.30 | 55.97 |
| | 30 | 56.20 | 73.37 | 64.78 |
| | 50 | 54.39 | 74.76 | 64.57 |
| | Avg. | 51.33 | 66.92 | 59.12 |

| Method | # Dem. | Level 1 | Level 2 | Macro |
|---|---|---|---|---|
| **Louvain; Default R; Smallest r** | 5 | 50.75 | 64.52 | 57.63 |
| | 10 | 48.92 | 60.77 | 54.85 |
| | 15 | 49.64 | 62.30 | 55.97 |
| | 30 | 58.82 | 75.62 | 67.22 |
| | 50 | 56.36 | 77.14 | 66.75 |
| | Avg. | 52.90 | 68.07 | 60.48 |
| **Louvain; Dense Graph; Smallest r** | 5 | 52.63 | 66.31 | 59.47 |
| | 10 | 47.06 | 60.87 | 53.96 |
| | 15 | 50.36 | 61.88 | 56.12 |
| | 30 | 53.66 | 71.07 | 62.36 |
| | 50 | 52.25 | 74.64 | 63.45 |
| | Avg. | 51.19 | 66.95 | 59.07 |
| **Louvain; Sparse Graph; Smallest r** | 5 | 48.57 | 60.00 | 54.29 |
| | 10 | 47.48 | 59.67 | 53.58 |
| | 15 | 49.64 | 62.30 | 55.97 |
| | 30 | 59.13 | 77.07 | 68.10 |
| | 50 | 56.36 | 77.14 | 66.75 |
| | Avg. | 52.24 | 67.24 | 59.74 |

Table 4.14: F1 scores for Random-based, KNN-based and graph-based approaches on the Privacy Policies dataset by balancing the labels in the prompt. The underlined configurations are the ones that perform better than both the Random and the KNN-based retrieval approaches in terms of average Macro F1 score.

game states and corresponding lists of legal moves. Chef's hat is a simple game with rule patterns that are easy to learn but not trivial, especially if no description of the rule is given in the prompt, making it a good candidate for this experiment. Moreover, the lack of literature about it with respect to other more famous board games such as Chess or Othello is expected to force the LLM not to rely on its pre-training data.

### 4.3.1 Dataset

The data has been obtained by letting 4 agents that always choose a random move play against each other for 100 matches. At each player's turn, we extracted the game state, which consists of a pair of board state and the player's hand, and the corresponding list of currently legal moves. After removing duplicates, this process resulted in 7819 unique triples of *Player Hands*, *Board States* and *Possible Actions*. Player's hands and board states are represented by lists of respectively 17 and 11 natural numbers ranging from 0 to 13, where each number corresponds to the value of a card, except for 0 and 12, which represent respectively a missing card and a Joker card. Legal moves are encoded as lists of elements of the form 'CX;QY;JZ',and*pass*, where the former represents the move which consists in playing Y cards with value X and Z Joker cards. Due to long inference time, we only use the 221 triples from the last 3 matches as a test set, reserving the remaining triples to the knowledge base.

### 4.3.2 Experimental Results

The highly structured format of the data allows us to experiment with different distances. Specifically, we use the *Hamming* distance between the concatenations of players' hands and board states. Moreover, we test the behaviour of the LLM both with symbolic demonstrations and with their respective verbalized versions.

**Example 4.3.1.** *Here is an example of a symbolic demonstration and its respective verbalized version:*

- ***Symbolic Demonstration:***

  *A: Player hand: (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 9, 9, 10, 10, 10, 11);*
  *Board state: (13, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)*
  *B: ['C9;Q1;J0', 'C9;Q2;J0', 'C10;Q1;J0', 'C10;Q2;J0', 'C10;Q3;J0', 'C11;Q1;J0', 'pass']*

- ***Verbalized Demonstration:***

  *A: Player hand: ['2 card(s) with value 9', '3 card(s) with value 10', '1 card(s) with value 11'];*
  *Board state: ['1 card(s) with value 13']*
  *B: ['play 1 card(s) with value 9', 'play 2 card(s) with value 9', 'play 1 card(s) with value 10', 'play 2 card(s) with value 10', 'play 3 card(s) with value 10', 'play 1 card(s) with value 11', 'pass']*

For evaluation we rely on the Intersection over Union (*IoU*) between the list of returned legal moves in a given game state and the actual list of legal moves in the same game state.

Table 4.15 and Table 4.16 show the IoU scores for the symbolic and the verbalized versions of the demonstrations, respectively. In all cases, using symbolic representations of the query and demonstrations leads to better results. For the symbolic representation, KNN performs better on average. However, it must be noted that the graph-based approach matches and surpasses the IoU scores achieved by KNN with multiple configurations, for large numbers of retrieved demonstrations. In fact, the graph-based approach has the largest improvement in performance when increasing the number of demonstration from 5 up to 50. On the contrary, the KNN-based method achieves

| | | **Number of Demonstrations** | | | | | |
|---|---|---|---|---|---|---|---|
| Method | | 5 | 10 | 15 | 30 | 50 | Average |
| **Random** | | 30.60 | 32.09 | 32.43 | 34.62 | 35.04 | 32.96 |
| **KNN** | | 43.71 | 43.49 | 43.65 | 44.25 | 43.95 | 43.81 |
| **Louvain** | **R=3; r=5** | 38.82 | 37.98 | 40.57 | 41.33 | 42.77 | 40.29 |
| | **R=3; r=6** | 38.00 | 37.81 | 38.72 | 40.10 | 41.20 | 39.17 |
| | **R=4; r=5** | 38.54 | 38.76 | 40.66 | 41.90 | 44.59 | 40.89 |
| | **R=4; r=6** | 37.94 | 37.71 | 38.64 | 39.34 | 42.88 | 39.30 |
| | **R=3; Smallest r** | 43.70 | 40.98 | 42.27 | 44.12 | **44.86** | 43.19 |
| | **R=4; Smallest r** | 42.89 | 41.51 | 41.76 | 42.34 | 43.92 | 42.48 |

Table 4.15: IoU scores for the Chef’s Hat board game with symbolic demonstrations.

| | | **Number of Demonstrations** | | | | | |
|---|---|---|---|---|---|---|---|
| Method | | 5 | 10 | 15 | 30 | 50 | Average |
| **Random** | | 32.15 | 32.61 | 33.27 | 33.33 | 32.40 | 32.75 |
| **KNN** | | 39.30 | 39.98 | 39.99 | 39.91 | 39.93 | 39.82 |
| **Louvain** | **R=3; r=5** | 37.66 | 38.26 | 38.02 | 37.94 | 38.02 | 37.98 |
| | **R=3; r=6** | 38.21 | **40.18** | 38.20 | 38.68 | 38.38 | 38.73 |
| | **R=4; r=5** | 37.85 | 38.57 | 39.18 | 38.59 | 39.54 | 38.75 |
| | **R=4; r=6** | 37.98 | 37.54 | 38.97 | 37.76 | 38.60 | 38.17 |
| | **R=3; Smallest r** | 39.87 | **40.18** | 38.63 | 38.78 | 38.84 | 39.26 |
| | **R=4; Smallest r** | 39.27 | 38.26 | 39.46 | 38.86 | 39.32 | 39.03 |

Table 4.16: IoU scores for the Chef’s Hat board game with verbalized demonstrations.

strong performance with even a small number of examples, improving only slightly as the number of demonstrations increases. This is likely due to the choice of the Hamming distance: in fact, demonstrations with pairs of players' hands and board states that are close to each other according to the Hamming distance (even with just a distance of 1) can admit extremely different lists of legal moves. This effect becomes even more relevant as the distance keeps increasing. Hence, when given few demonstrations, without having access to the rules of the game the model is not able to achieve good performances just by mimicking the demonstrations retrieved via the graph-based method. On the other hand, increasing the number of demonstrations helps it reconstruct abstract patterns and obtain better results.

## 4.4 Discussion

In most experiments, the KNN baseline proved to be the most competitive, showcasing the positive impact of similarity between demonstrations and query. On the other hand, the graph-based approaches are usually able to surpass both the Random and the Zero-Shot baselines. In most cases, the best average performances in terms of Macro F1 score has been achieved by one or more configurations of the graph-based methods, suggesting that this method is capable of retrieving effective demonstrations with appropriate settings. This, combined with the dramatic improvements obtained with label-balanced variants for classification tasks, confirms findings about the importance of diversity of demonstrations in the prompt. It is important to observe, however, that usually the KNN and graph-based approaches achieve similar scores, with the latter surpassing the former with only a few configurations. Moreover, among all configurations, the best performing ones are often those with a smaller radii. These observations show that in the experiments, the impact of relevance in the similarity-diversity tradeoff has been somewhat underestimated. Hence, we suggest that the radius should be set in general to lower values. In addition, the

10% increase in radius when there are not enough items in the knowledge base in the corresponding region of the embedding space is likely to be too large, partially explaining the non-monotonicity of the macro F1 scores, which in turn undermines the average scores of the graph-based approaches.

Conversely, the impact of the resolution parameter $R$ by itself on performances remains unclear given the results of the experiments. There are however a couple of meaningful observations concerning the interaction between the two parameters. First, decreasing $r$ can cause the relative performances of Dense and Sparse Demonstration Graphs to shift from Dense graphs performing better to Sparse graphs achieving better results, while the opposite shift almost never occurs. Second, again as $r$ decreases, the best performing configurations are in most cases ordered decreasingly by density. These two remarks suggest that the best performing values of radius and resolution are loosely positively correlated. This is intuitively justifiable following Remark 3.2.4. In fact, for extreme values $R = 0$ and $R \geq 2r$ we obtain respectively a totally disconnected subgraph and a clique, making the subsequent steps of community detection and node ranking ineffective. As $r$ decreases, keeping the resolution fixed pushes the extracted subgraph towards being a clique, losing exploitable local information.

The study of the interplay between radius and resolution, how to systematically set them, and how to appropriately assign the correct weight to similarity and diversity is left as future work.

# Chapter 5

# Conclusions

The main contribution of this thesis is the proposal of a fast, versatile and theoretically sound graph-based approach for demonstration retrieval in In-Context Learning. Previous studies indicate that similarity of examples to an input query and diversity of demonstrations are key factors in ICL performances. We exploit tools and notions from graph theory to choose examples that are relevant, semantically diverse and representative of concepts encoded in the knowledge base. Contrary to heuristic approaches based on maximizing of an abstract metrics (such as MMR-optimization), and learning-based methods, our approach naturally yields better interpretability. Similarly to KNN, which retrieves demonstrations that are relevant to the query because they are the most similar, we guide the LLM towards the correct output via demonstrations chosen for their capability to summarize diverse and relevant concepts.

To assess the effectiveness of our method, we conducted experiments on five different tasks, evaluating several configuration of the *radius* $r$ and *resolution* $R$ parameters and comparing results to Zero-Shot, Random and KNN baselines. In almost all cases the graph-based approach surpassed both the Zero-Shot and the Random baselines. The KNN baseline outperforms our approach in several configurations, but there are also cases in which the graph-based method performs better or on par with it.

The proposed method presents a huge design space to be explored. It is important to note that this work focuses much more on the theoretical motivations behind the outlined approach, rather than on finding the best possible configurations for the method itself. In fact, most decisions (such as the choice of parameters, centrality measures and graph-partitioning algorithms) were made based on some preliminary studies, due to lack of literature on the matter.

The most trivial direction for future work is to determine a systematic way to set the parameters $r$ and $R$. Another interesting topic to investigate is the impact of different community detection algorithms and other node centrality measures. Among the latters, the most promising ones are other spectral centralities such as the Katz and the subgraph centralities [19]. Additionally, we would like to assess the effectiveness of assigning positive weights (that decrease with distance in the knowledge base) to the edges of the Demonstration Graph. Lastly, we mention the possibility of working with generalized definitions of modularity that would allow to influence the number and the size of the communities detected via the Louvain algorithm.

# Bibliography

[1] P. Barros, A. Sciutti, A. C. Bloem, I. M. Hootsmans, L. M. Opheij, R. H. Toebosch, and E. Barakova. It's food fight! designing the chef's hat card game for affective-aware hri. In *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, HRI '21, pages 524–528. ACM, March 2021. DOI: 10.1145/3434074.3447227. URL: http://dx.doi.org/10.1145/3434074.3447227.

[2] P. V. A. Barros, A. C. Bloem, I. M. Hootsmans, L. M. Opheij, R. H. A. Toebosch, E. I. Barakova, and A. Sciutti. The chef's hat simulation environment for reinforcement-learning-based agents. *CoRR*, abs/2003.05861, 2020. arXiv: 2003.05861. URL: https://arxiv.org/abs/2003.05861.

[3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, October 2008. ISSN: 1742-5468. DOI: 10.1088/1742-5468/2008/10/p10008. URL: http://dx.doi.org/10.1088/1742-5468/2008/10/P10008.

[4] P. Bonacich. Power and centrality: a family of measures. *American journal of sociology*, 92(5):1170–1182, 1987.

[5] U. Brandes, D. Delling, M. Gaertler, R. Goerke, M. Hoefer, Z. Nikoloski, and D. Wagner. Maximizing modularity is hard, 2006. arXiv: physics/0608255 [physics.data-an]. URL: https://arxiv.org/abs/physics/0608255.

[6] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, 2008. DOI: 10.1109/TKDE.2007.190689.

[7] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. arXiv: 2005.14165. URL: https://arxiv.org/abs/2005.14165.

[8] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(6), December 2004. ISSN: 1550-2376. DOI: 10.1103/physreve.70.066111. URL: http://dx.doi.org/10.1103/PhysRevE.70.066111.

[9] Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, B. Chang, X. Sun, and Z. Sui. A survey on in-context learning. In Y. Al-Onaizan, M. Bansal, and Y. Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 1107–1128. Association for Computational Linguistics, 2024. URL: https://aclanthology.org/2024.emnlp-main.64.

[10] L. C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, 1978. ISSN: 0378-8733. DOI: https://doi.org/10.1016/0378-8733(78)90021-7. URL: https://www.sciencedirect.com/science/article/pii/0378873378900217.

[11] G. Grundler, A. Galassi, P. Santin, A. Fidelangeli, F. Galli, E. Palmieri, F. Lagioia, G. Sartor, and P. Torroni. AMELIA - argument mining evaluation on legal documents in italian: A CALAMITA challenge. In F. Dell'Orletta, A. Lenci, S. Montemagni, and R. Sprugnoli, editors, *Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024), Pisa, Italy, December 4-6, 2024*, volume 3878 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2024. URL: `https://ceur-ws.org/Vol-3878/124%5C_calamita%5C_long.pdf`.

[12] G. Grundler, R. Liepiņa, M. Musicco, F. Lagioia, A. Galassi, G. Sartor, and P. Torroni. Detecting vague clauses in privacy policies: the analysis of data categories using bert models and llms. In December 2024. ISBN: 9781643685625. DOI: `10.3233/FAIA241235`.

[13] G. Grundler, P. Santin, A. Galassi, F. Galli, F. Godano, F. Lagioia, E. Palmieri, F. Ruggeri, G. Sartor, and P. Torroni. Detecting arguments in CJEU decisions on fiscal state aid. In G. Lapesa, J. Schneider, Y. Jo, and S. Saha, editors, *Proceedings of the 9th Workshop on Argument Mining*, pages 143–157, Online and in Gyeongju, Republic of Korea. International Conference on Computational Linguistics, October 2022. URL: `https://aclanthology.org/2022.argmining-1.14/`.

[14] S. Gupta, M. Gardner, and S. Singh. Coverage-based example selection for in-context learning. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13924–13950, Singapore. Association for Computational Linguistics, December 2023. DOI: `10.18653/v1/2023.findings-emnlp.930`. URL: `https://aclanthology.org/2023.findings-emnlp.930/`.

[15] J. R. Kirkwood and B. H. Kirkwood. The perron–frobenius theorem. *Linear Algebra*, 2020. URL: `https://api.semanticscholar.org/CorpusID:120536925`.

[16] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen. What makes good in-context examples for GPT-3? In E. Agirre, M. Apidianaki, and I. Vulić, editors, *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics, May 2022. DOI: `10.18653/v1/2022.deelio-1.10`. URL: `https://aclanthology.org/2022.deelio-1.10/`.

[17] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9):195:1–195:35, 2023. DOI: `10.1145/3560815`. URL: `https://doi.org/10.1145/3560815`.

[18] 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. *CoRR*, abs/1907.11692, 2019. arXiv: `1907.11692`. URL: `http://arxiv.org/abs/1907.11692`.

[19] K. Mosler. Ernesto Estrada and Philip A. Knight (2015): A First Course in Network Theory, Oxford University Press, 272 pp., £29.99, ISBN 9780198726463. *Statistical Papers*, 58(4):1283–1284, December 2017. DOI: `10.1007/s00362-017-0961-1`. URL: `https://ideas.repec.org/a/spr/stpapr/v58y2017i4d10.1007_s00362-017-0961-1.html`.

[20] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2), February 2004. ISSN: 1550-2376. DOI: `10.1103/physreve.69.026113`. URL: `http://dx.doi.org/10.1103/PhysRevE.69.026113`.

[21] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report 1999-66,

Stanford InfoLab, November 1999. URL: http://ilpubs.stanford.edu:8090/422/. Previous number = SIDL-WP-1999-0120.

[22] N. Perra and S. Fortunato. Spectral centrality measures in complex networks. *Physical Review E*, 78(3), September 2008. ISSN: 1550-2376. DOI: 10.1103/physreve.78.036107. URL: http://dx.doi.org/10.1103/PhysRevE.78.036107.

[23] P. Santin, G. Grundler, A. Galassi, F. Galli, F. Lagioia, E. Palmieri, F. Ruggeri, G. Sartor, and P. Torroni. Argumentation structure prediction in cjeu decisions on fiscal state aid. In *ICAIL '23: 19th International Conference on Artificial Intelligence and Law*, Braga, Portugal. ACM, 2023.

[24] H. Su, J. Kasai, C. H. Wu, W. Shi, T. Wang, J. Xin, R. Zhang, M. Ostendorf, L. Zettlemoyer, N. A. Smith, and T. Yu. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL: https://openreview.net/forum?id=qY1hlv7gwg.

[25] X. Wang, J. Wu, Y. Yuan, M. Li, D. Cai, and W. Jia. Demonstration selection for in-context learning via reinforcement learning. *CoRR*, abs/2412.03966, 2024. DOI: 10.48550/ARXIV.2412.03966. arXiv: 2412.03966. URL: https://doi.org/10.48550/arXiv.2412.03966.

[26] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022, 2022. URL: https://openreview.net/forum?id=yzkSU5zdwD.

[27] J. Ye, Z. Wu, J. Feng, T. Yu, and L. Kong. Compositional exemplars for in-context learning. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *International Conference on Machine*

*Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 39818–39833. PMLR, 2023. URL: `https://proceedings.mlr.press/v202/ye23c.html`.

[28] X. Ye, S. Iyer, A. Celikyilmaz, V. Stoyanov, G. Durrett, and R. Pasunuru. Complementary explanations for effective in-context learning. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4469–4484, Toronto, Canada. Association for Computational Linguistics, July 2023. DOI: `10.18653/v1/2023.findings-acl.273`. URL: `https://aclanthology.org/2023.findings-acl.273/`.

[29] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J. Nie, and J. Wen. A survey of large language models. *CoRR*, abs/2303.18223, 2023. DOI: `10.48550/ARXIV.2303.18223`. arXiv: `2303.18223`. URL: `https://doi.org/10.48550/arXiv.2303.18223`.

# Appendices

# Appendix A

# Prompt Templates

In this Appendix we show the prompt templates that have been used in all experiments. The templates for the Demosthenes corpus and the privacy policies dataset have been directly taken and/or adapted from [11] and the repository of [12], respectively.

## A.1 Argument Mining

Argument Classification:

> *Classify the following argumentative text as premise 'prem' or conclusion 'conc'. A premise (prem) is a proposition that provides a reason or support for the argument. A conclusion (conc) is the statement that follows logically from the premise(s) and represents the final point being argued for. Only reply with 'prem' or 'conc'.\nExamples:*

Type Classification:

> *Classify the following premise as factual 'F', legal 'L' or both. Factual premises (F) describe factual situations and events, pertaining to the substance or the procedure of the case. Legal premises*

> *(L) specify the legal content (legal rules, precedents, interpretation of applicable laws and principles). The expected output is a list with all applicable labels. For example: ['F', 'L']. Only reply with the list of labels.*

Scheme Classification:

> *Classify the following legal premise as one or more of the following argumentative schemes: Rule, Prec, Class, Itpr, Aut. Rule: whether there is an explicit or implicit reference to an article of law or citation of the text of a certain article. Prec: whether there is a reference to a previous ruling of the Supreme Court or the Court of Justice of the European Union. Class: if there is a definition of a legal concept or its constituent elements. Itpr: if there is reference to one of the interpretative criteria contained in Article 12 of the prelegislations (literal, teleological, psychological, systematic) to the Civil Code. Aut: if there is a reference to an indication by an authority (e.g. an opinion of the Advocate General). The expected output is a list with all applicable labels. For example: ['Prec', 'Aut', 'Rule']. Only reply with the list of labels.*

## A.2 Privacy Policy Compliance

> *You will be given as input a sentence from a privacy policy that contains information about what data the service collects about the user. You have to classify the sentence into one of the following classes: "sufficiently informative" or "insufficiently informative".*
>
> *In doing so, consider that GDPR, and the EDPB's Guidelines,*

*contain a certain inherent tension, namely that between requiring that the information is provided in as easy a way to understand as possible (comprehensibility) and that it is concrete and definitely (comprehensiveness).*
*Sometimes, using open-ended qualifiers like "for example" or "such as" might actually facilitate understanding by the data subject, especially when terms not often used in the natural language (e.g. "device information" or "geolocation information") are concerned. For this reason, we differentiate between:*
*Abstract terms (e.g. usage information) vs. concrete terms (e.g. geolocation information)*
*Open-ended qualifications ("for example", "such as", etc.) vs. closed-catalogues ("meaning", "understood as" ).*

*A sentence containing an abstract term, UNLESS followed by a comprehensive enumeration, should be judged as "insufficiently informative".*
*A sentence containing a concrete term, even if followed by an open-ended qualifier, should be judged as "sufficiently informative".*

*Avoid explanations. Only reply with "insufficiently informative" or "sufficiently informative".*

*Here are some examples:*

## A.3 Legal Moves Generation

*You are an expert board game player, and you are playing a card game called Chef's Hat. This game is played with cards with values ranging from 1 to 13, both included.*

*In the following examples "A" provides two lists.*
*The first list is a description of the cards in hand of a player.*
*The second list is a description of the cards that are already on the board.*

*Then "B" replies with the full list of the current legal moves in the situation presented by "A".*

*Complete the text by listing all the current legal moves given the last description from "A". Only add the text that comes after "B:".*

# Acknowledgements

It has been quite a journey, starting in the middle of covid as a Mathematics graduate bachelor who did not even know what a csv was. It has been challenging, especially at the beginning, but it has been mostly an interesting and exciting challenge, rather than an oppressing one.

My first acknowledgements go to prof. Torroni, who has never failed to help me, starting from the admission procedure up to the last project work, being open to and even encouraging the skills and interests that come from my previous formation. I cannot help but thank my advisors Prof. Galassi and Dr. Grundler for their astonishing availability and for their precise and most valuable guidance, both in conducting experiments and in the writing process, which helped me to do the best possible work for my thesis. I truly could not have asked for anything better.

I thank my friends from my time in Udine, especially Matteo, Nikola and Stefano, for helping to make my experience there much less hideous, and for continuing to share good times with me afterwards, even if we are spatially distant.

I thank my passed away grandmother who is largely responsible for inspiring me to pursue this path and ultimately write this thesis, and the other grandparents who always celebrate whenever I share my small successes with them.

I thank my parents for their unwavering moral support and affection and my brothers for always making me feel at home, no matter where I am.

Of course, I thank all the rest of my family and all the people that love me,

for their constant support and for always wishing me the best.

Finally, a special thanks to Thuvarakah, the best partner I could even imagine. You were always there in my darkest moments and in sharing joys, and have always made me feel more confident. I am only including your contribution to this path, but a complete acknowledgement would more than double the pages in this thesis. I love you.