Graph-based approaches for Demonstration selection in the In-Context Learning setting 3-cfu Project Work

Francesco Alfieri
Master's Degree in Artificial Intelligence, University of Bologna
francesco.alfieri5@studio.unibo.it

Abstract

This project focuses on investigating the capabilities of LLMs to assess the legality of moves of a relatively obscure board games (Chef's Hat) via ICL. Specifically, we focus on the demonstration selection aspect of ICL, with a flexible and generalizable graph-based method. The goal is to retrieve demonstrations that are relevant to a given query and able to summarize diverse concepts from the available examples in the knowledge base. After creating an ad-hoc dataset of pairs of players' hands, board states and legal moves, we test different configurations against a random and a KNN baselines, highlighting both promising results and clear limitations.

1 Introduction

Within this project, we perform experiments testing the capabilities of LLMs to understand the rules of a simple board game via in-context learning (ICL). It has been observed that the performances of ICL is sensitive to many settings, including prompt templates, selection and order of demonstration examples (Wang et al., 2024). In particular, we focus on the demonstration examples selection, comparing two baselines (random and KNN) to graph-based approaches. The goals of these kind of methods is manifold: we aim to select examples that are semantically relevant to a given query and that are able to summarize concepts that are meaningful and diverse.

The specific game is called Chef's Hat, a four-players competitive card game (Barros et al., 2021). The games are divided into three rounds: the first round involves at most a couple decisions, the second round is the one in which the majority of the game and decisions take place, and the last one is a cleanup phase where the players have no agency. We attach the link to the rulebook in Section 8.

The choice of the game brings two advantages. First, it is very simple, but not *too* simple: the patterns of the rules are easy to learn, but not trivial, especially where no description of the rules is given in the prompt. Second, it is not a widely known game, so that LLMs cannot exploit too much of its training data contrary, for example, to chess, for which there is a lot of literature available.

We focus specifically on the second round of the game. In this phase, players have two options when taking turns: they can either place *ingredient* cards, numbered from 1 to 11 and two Joker cards, on the board (thematically named *pizza*), or pass the turn. The played cards need to have lower face values than the previously played cards. Players can play multiple copies of a card at once, but always have to play an equal or greater amount of copies than the previous player did. If a player cannot (or does not want to) play cards, they pass until the next *pizza* starts (Barros et al., 2020).

2 Background

In this project we focus on demonstration selection for ICL. It has been shown that choosing close (usually with high cosine similarity in the embedding space) examples that are semantically relevant to test instances via KNN can greatly improve the performance of ICL (Liu et al., 2022). In addition to similarity, diversity of demonstrations has been reported to play a key role both within the prompt and for the pruposes of selective annotation for low budget frameworks with unlabeled data. (Wang et al., 2024), (Su et al., 2022). Specifically, in (Su et al., 2022) a graph-based approach is adopted in order to sample diverse candidate demonstrations from a pool of unlabeled data in advance, independently from any specific test instance. It has been observed that diversity is more important for difficult tasks, such as NLI and semantic parsing, but relevance is more crucial for simpler tasks such as sentiment analysis (Ye et al., 2023).

Here, we employ a somewhat similar graph-based approach in order to make use of the full labeled dataset, so to extract demonstrations that **(i)** are relevant to a given query, **(ii)** make for good summaries of the semantic contents of other discarded examples, and possibly **(iii)** are diverse to other examples provided within the prompt.

In the following we use various tools from graph theory, with the most important ones being the PageRank scoring algorithm (Page et al., 1999) and the Louvain method for community detection (Blondel et al., 2008).

Usually, spectral centrality are based on the principle that the importance of a node depends on the importance of its neighbors (Perra and Fortunato, 2008). Importantly, in this context, where edges represent similarity between demonstrations, examples are deemed *important* if they are similar to nodes that are the most similar to other examples, and this is the reason we will choose them as good summaries/representatives of concepts embedded in subgraphs and communities. The most straightforward spectral measure is probably the eigenvector centrality:

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

where A is the adjacency matrix of the graph, and λ is chosen so to normalize the vector of scores $\mathbf{v}$. In general, however, there can be many different nontrivial eigenvalues λ for which a non-zero eigenvector solution exists. For this reason, we use instead the PageRank algorithm in order to score nodes. This is the eigenvector centrality of the weighted graph associated to a convex linear combination of the transition matrix N of the original graph and the transition matrix of a random jump:

$$\lambda \mathbf{p} = G^T \mathbf{p},$$

where

$$G = \alpha N + (1-\alpha)\mathbb{1}^{n\times n},$$

with $N = \text{diag}(A\mathbb{1})^{-1}A$, n the number of nodes in the graph, and $0 < \alpha < 1$, usually set to $\alpha = 0.85$. The PageRank vector of scores is defined as the normalized eigenvector associated to the largest positive eigenvalue λ, and its well-definedness is a direct consequence of the Perron-Frobenius theorem, whose application is made possible by the inclusion of the random jump.

The Louvain method is a fast heuristic process which aims at maximizing the modularity of partitions of graphs. In this context, we use it to detect clusters of nodes that are similar to each other and are possibly related to the same concept. Modularity is a measure of the quality of graph partitions, which compares the ratio of intra-community edges in the partition with the same expected value in random graphs with the same degree distribution. This is an iterative process which repeatedly move each node from a partition to another neighboring one, and then building new networks that have as nodes the communities identified in the previous step. For a thorough explanation cfr. (Blondel et al., 2008). On important aspect to note is that this method depends on the order in which nodes are considered. For this reason, we run tests on 3 different seeds when using this method.

3 System description

In all experiments we tested the various methods for example selection against the same test set of 221 couples of *player hands* and *board state*.

In all cases the employed LLM is **Llama-3.1-8B-Instruct** (`https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct`), with 4 bit quantization in order to reduce computational and memory costs. We use the text generation pipeline from the transformers library with default parameters except for *max_new_tokens*, which has been manually set to 500 in order to be able to fit the longest sequences of legal moves from the dataset.

The first approach, which acts as the simplest baseline, is random selection. In this case the demonstrations are simply randomly sampled from the available ones.

The second approach is KNN. For each test instance, we simply select the K closest examples for in-context learning. The "closest" examples are often selected according similarity (e.g. cosine similarity) between sentence embeddings. In this case, however, given the highly regular structure of the inputs and expected outputs of the model, we rely on the *Hamming distance* between concatenations of sequences of *Player_Hands* and *Board_Before*.

Last we include the graph-based approaches. In all cases we first construct the *Demonstration graph*. This generation is determined only by a single parameter R, which we call *resolution*. Each instance in the knowledge base is represented by a node, and for each pair of nodes, they are connected by and edge if and only if the Hamming distance between the respective concatenations of

Player_Hands and *Board_Before* is less than or equal to the resolution parameter. As a consequence, the lower the resolution, the sparser the resulting graph becomes.

After the examples are retrieved, they are joined and included in templates of increasing informative content and complexity: the first one does not contain any information about the semantics of the input and rules of the game, and only consists in an instruction to complete the text based on the given demonstrations; the second template includes informations about the task and what the numbers and input/output lists represent, still with no indication concerning the specific rules of the game; the last one adds an explanation of the rules of this phase of the game.

3.1 Fixed-radius subgraphs

In this case at test time, for each test instance, we extract a tailored subgraph of the *Demonstration graph* by removing all nodes that represents examples with a Hamming distance from the test instance larger than a parameter (*radius*) r. Last, the 5/10/15 nodes with the highest pagerank score **in the subgraph** are selected and the corresponding demonstrations are fed to the LLM. If the set radius yields a subgraph with less han 5/10/15 nodes, than the smallest radius that yields a sufficiently large subgraph is selected. It is important to note that in general the radius should be at least greater than half the resolution. In fact, due to the triangular inequality, if the resolution parameter is more than double the radius, then the extracted subgraph is a clique, and any centrality measure assigns to all nodes in permutation-invariant graphs the same score.

3.2 Smallest-radius subgraphs

Since preliminary observations showed that the previous method was not able to match the performance of KNN selection, we also tested getting rid of the radius parameter. This approach is analogous to the previous one, except that the size of the subgraph is not fixed *a priori*, but depends on how many demonstrations close to the test instance are present in the knowledge base. In this case, we select the smallest radius which allows the demonstration subgraph to contain at least 5/10/15 nodes. In this way, we try to mediate the theoretical benefits from graph theory with the experimental ones from KNN which suggest that selecting very close examples to test instances is crucial to achieve good performances. This can be achieved due to the fact that usually many nodes lie at exactly the same distance from the test instance due to the discrete nature of the Hamming distance. In more generic settings, where this method would collapse to KNN, this can be approximated by setting a very low radius and repeatedly increasing it by small, fixed amounts.

3.3 Variants

We also tested two variants of the former two methods. The first one consists in applying weights to edges in the demonstration graph which decay when the Hamming distance between the connected nodes increases. Specifically, we tested assigning weights $w(i, j) = \frac{1}{n}$ to edges (i, j), if and only if $d_{Hamming}(i, j) = n$.

The second variation is an application of the Louvain method for community detection in the extracted subgraph. In this case we run the Louvain method and select the top node (pagerank-wise **in the respective community**) from the 5/10/15 most numerous communities identified. If not enough communities are detected, then the top k nodes from the most numerous communities are selected, where k is the smallest natural number which allows to select enough demonstrations. This is the method that allows the examples to be *diverse*. In fact in this case the three requirements from Section 2 are satisfied: **(i)** the examples are relevant because they are chosen from a selection of examples close to the query; **(iii)** they are representative of diverse concepts as they belong to disjunct and large communities and; **(ii)** they make for *good* representatives of those concepts because they are selected with the pagerank metric. A downside of this approach is that the Louvain method does not allow to set the amount of generated partitions beforehand. Instead, it relies on the resolution parameter γ (set by default on 1) which favors larger communities if less than 1, and smaller communities otherwise.

4 Data

The data have been obtained using a python script from the Chef's Hat GYM repository, linked in Section 8. In this case, four agents that make random choices at each of their turns have been set to play against each other for 100 matches. At each action (both taken from players or state-based) the lists of cards in the players' hands are updated, together with the states of the board (both before and after the play), and the list of actions available in the current situation. Since we focus on the second phase of the game, we extract from the obtained snapshots of the matches only the triples of *Player_Hands*, *Board_Before* and *Possible_Actions* where the *Action_Type* is "Discard".

After removing duplicates, this process results in 7819 unique triples of *Player_Hands*, *Board_Before* and *Possible_Actions*. Due to long inference time, we only use the 221 triples from the last 3 matches to evaluate the performances of each approach, reserving triples coming from the first 97 matches as demonstrations to be given to the LLM. This split with much less test instances than *"train"* instances has been chosen merely in order to reduce the time required for testing the various methods.

Players' hands and boards 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. The legal moves are encoded as lists of elements of the form CX;QY;JZ and *pass*, where the former type of elements represents the move which consists in playing Y cards with value X and Z Joker cards.

5 Experimental setup and results

In all experiments we compared the performances on the test set of various techniques for example selection. In order to do so, we compare several metrics that measure the capabilities of the LLM of correctly identifying legal moves and disregarding illegal ones.

The main metric used to compare the performances is the *intersection over union*: this measures the ratio between the number of correctly identified legal moves and the number of elements in the union of all actual legal moves (both identified and not identified by the LLM) and the illegal moves returned by the LLM. Other metrics that allow a better understanding of the effects of the methods are the *all moves accuracy* (the fraction of test instances for which the IoU is 1), and the separate counts of *true positives* (TP), *false positives* (FP, moves that are returned by the LLM but are not legal), and *false negatives* (FN, moves that are legal but are not returned by the LLM). An upside of using the *intersection over union* over the accuracy on the entire lists of legal moves is that the former is much more granular and allows to appreciate smaller differences in performances (both increases and decreases) by accounting for both kinds of errors and correct moves predicted, while the latter, especially given the limited size of the test set, is not able to give meaningful insight about the behaviour of different approaches. Moreover, IoU is a much more stable measure with respect to the FP count. In fact, often in difficult cases the LLM keeps generating illegal moves which can increase the FP count by a lot (up to 50 per test instance), and in those cases the IoU score is simply 0. Despite the great variability of FP counts, they are still an important metric as the vast majority of errors is of this type, and reducing the FP counts is one of the priorities for improving these approaches. Note that we do not include here tables for all metrics and experiments due to the lack of dedicated space but they are included in the respective notebooks.

In all cases 5/10/15 examples are given with 3 different prompts of increasing complexity, included in the appendix, for a total of 9 combinations per run test. The parameters to be set are the resolution R of the demonstration graph, the radius r of the extracted subgraphs and the resolution parameter γ for the Louvain method, left to the default of 1. Various configurations have been tested, with R ranging from 3 to 5, and r assuming values in $\{6, 8, 10\}$. The maximum radius has been manually set so that every test instance has at least 15 examples in the respective subgraph. However, this radius proved to be too large for most test instances and smaller values have been subsequently tested. On the other hand R has been chosen in a way that mantains the sparsity of the network, so to make use more of local informations of graphs. Where the chosen examples depend on random factors (that is, in the random and KNN methods, as well as Louvain's), we select three fixed seeds and report the averages of the resulting metrics.

Tables 1 through 9 report the IoU scores, TP and FP counts for all the used methods.

	Minimal template			*Intermediate template*			*Complex Template*		
# of examples	5	10	15	5	10	15	5	10	15
Random baseline	.259	.205	.265	.259	.277	.312	.163	.201	.240
KNN baseline	.431	.441	.462	.434	.470	.486	.270	.374	.408

Table 1: IoU scores of the two baselines.

	Minimal template			*Intermediate template*			*Complex Template*		
# of examples	5	10	15	5	10	15	5	10	15
R=5; r=10	.271	.300	.329	.284	.309	.309	.208	.259	.278
R=5; r=6	.376	.392	.395	.366	**.405**	.394	.275	.321	.338
R=4; r=8	.326	.324	.357	.300	.328	.350	.226	.306	.313
R=4; r=6	.365	.378	.368	.352	.387	.382	**.299**	**.338**	.355
R=3; r=8	.308	.322	.354	.315	.342	.351	.220	.287	.315
R=3; r=6	.347	.362	.390	.330	.366	.396	.272	.315	.340
Louvain; R=5; r=6	**.379**	**.398**	**.401**	**.384**	.400	**.416**	.256	.333	**.376**
weighted; R=5, r=6	.370	.384	.385	.347	.385	.376	.265	.331	.353

Table 2: IoU scores for fixed-radius approaches with various parameter configurations. Results suggests that subgraphs with a smaller radius r behave generally better. On the other hand, denser demonstration graphs with higher resolution R achieve better performance. The Louvain method outperforms all other approaches in most cases.

		Minimal template			*Intermediate template*			*Complex Template*		
# of examples		5	10	15	5	10	15	5	10	15
R=3		**.417**	.425	**.446**	**.431**	**.447**	**.466**	.259	.351	.391
R=5		.404	.402	.425	.421	**.447**	.456	**.284**	.350	.380
weighted; R=3		.406	.419	.437	.405	.421	.455	.244	.345	**.394**
weighted; R=5		.392	.416	.435	.376	.404	.449	.230	.337	.363
Louvain; R=3		.415	**.444**	.423	.422	.419	.434	.261	.332	.373
Louvain; R=5		.399	.404	.409	.424	.442	.452	.249	**.354**	.380

Table 3: IoU scores for smallest-radius subgraph approaches with different values for Resolution R. Contrary to the observations from Table 2, it seems that for very small subgraphs, sparser demonstration graphs prove more effective.

	Minimal template			*Intermediate template*			*Complex Template*		
# of examples	5	10	15	5	10	15	5	10	15
Random baseline	653	592	742	645	714	803	688	705	819
KNN baseline	828	856	874	835	864	869	688	829	830

Table 4: TP counts of the two baselines

	Minimal template			*Intermediate template*			*Complex Template*		
# of examples	5	10	15	5	10	15	5	10	15
R=5; r=10	708	769	793	800	823	788	728	782	799
R=5; r=6	800	**840**	**865**	**854**	880	856	755	831	818
R=4; r=8	784	764	821	803	837	834	720	799	833
R=4; r=6	774	816	824	825	856	837	746	**832**	**851**
R=3; r=8	755	776	811	810	834	826	703	780	804
R=3; r=6	773	812	844	803	**882**	848	**756**	809	820
Louvain; R=5; r=6	**810**	838	853	841	850	859	682	752	823
weighted; R=5; r=6	828	810	863	836	879	**867**	725	823	841

Table 5: TP counts for fixed-radius approaches with various parameter configurations.

		Minimal template			*Intermediate template*			*Complex Template*		
# of examples		5	10	15	5	10	15	5	10	15
R=3		802	851	855	823	**876**	854	649	**825**	821
R=5		789	844	**870**	850	864	**871**	**721**	822	**849**
weighted; R=3		781	**852**	848	802	859	867	656	815	834
weighted; R=5		775	828	859	793	857	851	622	803	773
Louvain; R=3		**822**	827	843	**852**	834	847	683	799	836
Louvain; R=5		796	838	838	827	843	850	667	821	830

Table 6: TP counts for smallest-radius demonstration graphs approaches with different values for Resolution.

	Minimal template			*Intermediate template*			*Complex Template*		
# of examples	5	10	15	5	10	15	5	10	15
Random baseline	2679	6485	4032	1716	2939	2085	6903	4855	4975
KNN baseline	1114	1209	1180	1171	1122	1094	3312	2083	1668

Table 7: FP counts of the two baselines

	Minimal template			*Intermediate template*			*Complex Template*		
# of examples	5	10	15	5	10	15	5	10	15
R=5; r=10	2128	2109	1617	2679	2158	1996	4622	3519	2902
R=5; r=6	1551	**1404**	1477	1858	1546	1503	3544	2849	2417
R=4; r=8	1708	1670	1553	2385	2124	1852	4855	2754	2342
R=4; r=6	1517	1410	1510	1922	**1478**	1617	**3434**	**2218**	2040
R=3; r=8	1806	1602	1482	2237	2115	1757	4171	2742	2435
R=3; r=6	**1457**	1642	**1378**	2042	1705	**1407**	3616	2373	2047
Louvain; R=5; r=6	1704	1435	1452	**1643**	1554	1448	3717	2461	**1834**
weighted; R=5; r=6	1472	1447	1551	1775	1625	1787	3877	2632	2095

Table 8: FP counts for fixed-radius approaches with various parameter configurations.

		Minimal template			*Intermediate template*			*Complex Template*		
# of examples		5	10	15	5	10	15	5	10	15
R=3		1180	1358	**1316**	**1201**	1239	**1223**	**3255**	2319	1914
R=5		**1160**	**1266**	1396	1431	1242	1264	3386	**2294**	**1858**
weighted; R=3		1186	1387	1321	1317	1291	1266	3652	2536	1906
weighted; R=5		1186	1383	1358	1439	1372	1355	3678	2594	1973
Louvain; R=3		1257	1322	1350	1361	1281	1314	3381	2357	1945
Louvain; R=5		1188	1412	1338	1292	**1223**	1320	3260	2427	1873

Table 9: FP counts for smallest-radius demonstration graphs approaches with different values for Resolution.

6 Discussion

Tables from the previous section show that in all cases the *intermediate* template achieves better performances than the minimal one. This highlights the fact that appropriate prompts are crucial for improving ICL performances.On the contrary, the most complex template with a complete description of rules yields a severe degradation of metrics. This is probably due to the fact that in this last case the inputs become much longer, on top of requiring more complex reasoning, such as variable substitution, due to how the rules are formulated (see Appendix). It is likely that larger LLM can obtain better results with these kind of templates with respect to the previous two.

Another immediate observation is that in almost all cases the performances improve with respect to the number of demonstrations provided in the prompt. This trend suggests that further increasing the number of examples might prove beneficial.

Tables 4 through 6 show that generally the approaches have similar behaviour with regard to the ability of correctly recognizing legal moves in some cases even surpassing the KNN baseline. On the contrary, Tables 7 through 9 show that the vast majority of errors come from returned moves that are not legal (the count of this kind of errors is usually one order of magnitude greater than the count of "*false negative*" moves). Moreover, the counts of false positives vary greatly and do not provide definitive insights on the behavior of the different methods.

Concerning the main considered metric (IoU), all of the proposed methods greatly surpass the random-selection baseline, but fall short compared to the KNN-selection one. Among the fixed-radius approaches, the Louvain method performs the best, suggesting that improving the diversity of examples which are not necessarily too close to the test instances can be beneficial. Smallest-radius subgraphs confirm that the distance of examples from test instances is the most impactful factor for achieving better results. In fact, all of these methods outperform the fixed-radius subgraphs approaches (with very few exceptions), which in most cases can select demonstrations that are further from the test instances. In these cases, the Louvain method does not perform better than the standard approach. A reasonable explanation is that since in most cases the selected demonstrations are extremely close to the test instances, and that in such cases the parameter R can be larger than or close to double the parameter r, the extracted subgraph can result in an extremely dense central community, from which the first example is extracted, and a number of peripheral and further communities, from which subsequent demonstrations are selected.

The extremely large impact of distance on performances is probably due to the specific choice of distance between examples. In fact, demonstrations with pairs of players' hands and board states that are close to each other according to the Hamming distance can admit extremely different lists of legal moves. For instance, a player with a hand of (0, 0, 0, 0, 6, 7, 7, 8, 8, 9, 9, 9, 10, 10, 11, 11, 11) has exactly 11 legal moves on the board (11, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) including *pass*, 6 legal moves on the board (11, 11, 0, 0, 0, 0, 0, 0, 0, 0, 0), and 1 legal move (just *pass*) on the board (6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), even the latter two game states have a Hamming distance of just 1 from the first one. On the other hand, this hand admits the same set of legal moves on boards (11, 11, 11, 0, 0, 0, 0, 0, 0, 0, 0) and (10, 10, 10, 0, 0, 0, 0, 0, 0, 0, 0), which are at Hamming distance 3 from each other. Since a single difference in the lists of elements is enough to produce many changes in the list of legal moves, there are two main consequences:

1. The performances of ICL degrade very fast with respect to small increments of the Hamming distance;
2. The performances of ICL with demonstrations very close to the test instances are more likely to be better due to a "lucky" mimicking of the outputs of the provided examples rather than a better understanding of the patterns and rules of the game.

This second claim is supported by the observation that, while in general the LLM shows a relatively good capability of generating moves based on their hand and mediocre capabilities in discarding moves based on the current board state, in some instances it generates moves that are illegal even just based on the player's hand due to similarity to provided demonstrations. For instance (from a test with the fixed-radius, weighted subgraph approach), given the query

> A: (0, 4, 5, 6, 7, 7, 8, 8, 8, 9, 9, 9, 10, 10, 10, 10, 11), (13, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
> B:

, and demonstration, among others

> A: (0, 0, 5, 6, 7, 7, 7, 8, 9, 9, 9, 10, 10, 10, 11, 11, 11), (13, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)
> B: ['C5;Q1;J0', 'C6;Q1;J0', 'C7;Q1;J0', 'C7;Q2;J0', 'C7;Q3;J0', 'C8;Q1;J0', 'C9;Q1;J0', 'C9;Q2;J0', 'C9;Q3;J0', 'C10;Q1;J0', 'C10;Q2;J0', 'C10;Q3;J0', 'C11;Q1;J0', 'C11;Q2;J0', 'C11;Q3;J0', 'pass']

, the model returns in the list of legal moves the following:

> 'C11;Q2;J0', 'C11;Q3;J0'

, which can only be due to a naive imitation of the demonstration.

The observations about the performances of ICL in the fixed-radius subgraph approach and the considerations about the great impact of the Hamming distance suggest that these kind of methods can prove beneficial given one of the following:

- They are applied in more generic contexts, such as Question Answering, where distances between sentence embeddings are better representatives of the semantics of the inputs;

- A more relevant distance, able to highlight substantial differences in the game state, is used.

7 Conclusion

In this project we explored the capabilities of LLMs in playing relatively unknown board games through ICL, with a special focus on demonstration selection. Inspired by the work of (Su et al., 2022) we developed a graph-based method for example retrieval, aiming to follow the principles of relevance, diversity and relative importance.

Observations from approaches where the candidate demonstrations are chosen from a larger pool, with elements with greater Hamming distance from testing instances, suggest that local informations and graph clustering can have a positive impact on ICL performances.

However, in this case, such improvements are clearly largely overshadowed by the great impact that larger Hamming distances have on the degradation of performances. As a consequence, in this specific settings the approach fails in achieving the same performances of KNN-based demonstration retrieval and alternatives should be explored. We believe that this method can be extended in different settings where small distances between demonstrations actually encode stronger semantic similarities, while in this specific setting, more sophisticated similarity metrics are clearly needed.

Furthermore, the flexible approach we presented comes with an extremely large design space that can be explored. Namely, a non-exhaustive list of topics that can be further investigated is:

- The designated metric to measure (dis)similarity between demonstrations and test instances;

- The choice of the measure of importance of nodes in subgraphs in place of the PageRank centrality score;

- Systematic ways to set the parameters R and r;

- For weighted approaches, a different degradation method of the weight of edges with respect to the distance/dissimilarity;

- For the Louvain method, a systematic way to choose the resolution parameter that influences the number of detected communities;

- Alternatives to the Louvain method for (potentially overlapping) community detection;

- Different combinations of the presented and suggested methods from this report;

- The use of more powerful and larger LLMs.

8 Links to external resources

- Chef's Hat repository: `https://github.com/pablovin/ChefsHatGYM?tab=readme-ov-file`

- Chef's Hat Rulebook: `https://github.com/pablovin/ChefsHatGYM/blob/master/gitImages/RulebookMenuv08.pdf`

References

Pablo Barros, Anne C. Bloem, Inge M. Hootsmans, Lena M. Opheij, Romain H. A. Toebosch, Emilia Barakova, and Alessandra Sciutti. 2020. The chef's hat simulation environment for reinforcement-learning-based agents.

Pablo Barros, Alessandra Sciutti, Anne C. Bloem, Inge M. Hootsmans, Lena M. Opheij, Romain H.A. Toebosch, and Emilia Barakova. 2021. 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, page 524–528. ACM.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *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.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

Nicola Perra and Santo Fortunato. 2008. Spectral centrality measures in complex networks. *Physical Review E*, 78(3).

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2022. Selective annotation makes language models better few-shot learners.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6).

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023. Compositional exemplars for in-context learning.

A Appendix - Prompt templates

- Base template:

  > “Complete the following text. Only add the text that comes after "B:"”

- Intermediate template:

  > “In the following examples "A" provides two lists. The first list is a description of the cards in hand of a board game player. The second list is a description of the cards that are already on the board. In both cases a "0" represents a missing card. Then, "B" replies with the full list of the current legal moves according to the description of "A". Complete the text by listing the correct legal moves given the last description from "A". Only add the text that comes after "B":”

- Complex template with full description of the rules:

  > “You are an expert board game player, and you are playing a new card game called Chef’s Hat. 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. In both cases a "0" represents a missing card, while a "12" represents a Joker card. "B" replies with the full list of the current legal moves according to the description of "A". The move ’CX;QY;JZ’ means that Y cards with value X are played, while JZ indicates that Z Jokers are played.\n As a player, you must play cards on top of all the cards that have been placed on the board before. The cards played must all have the same value, and this value must be strictly lower than the value of cards currently on the board. You cannot place less cards on the board than the amount of cards currently present there. If you have Joker cards, you can use them as if they had any value.\n Complete the text by listing all the current legal moves given the last description from "A". Only add the text that comes after "B":”