Graph-based approaches for Demonstration selection in In-Context Learning

Comparative experiments with the Chef's Hat card game

Francesco Alfieri
francesco.alfieri5@studio.unibo.it

28/01/2025

Introduction

We perform experiments about the capabilities of LLMs to understand the legality of moves in board games via In-Context Learning, specifically focusing on Demonstration Selection.

The specific board game we focus on is *Chef's Hat* (https://whisperproject.eu/chefshat):

Introduction

We perform experiments about the capabilities of LLMs to understand the legality of moves in board games via In-Context Learning, specifically focusing on Demonstration Selection.

The specific board game we focus on is *Chef's Hat* (https://whisperproject.eu/chefshat):

Two main advantages:

Introduction

We perform experiments about the capabilities of LLMs to understand the legality of moves in board games via In-Context Learning, specifically focusing on Demonstration Selection.

The specific board game we focus on is *Chef's Hat* (https://whisperproject.eu/chefshat):

Two main advantages:

- Simple game;

Introduction

We perform experiments about the capabilities of LLMs to understand the legality of moves in board games via In-Context Learning, specifically focusing on Demonstration Selection.

The specific board game we focus on is *Chef's Hat* (https://whisperproject.eu/chefshat):

Two main advantages:

- Simple game;
- Lack of literature with respect to more famous games (e.g. Chess, Othello).

Chef's Hat

Chef's Hat

There are three phases per round (*Shift*):

Chef's Hat

There are three phases per round (*Shift*):

1. Start of the Shift: cards and roles are dealt to the players. Players exchange up to 2 cards with each other depending on their role;

Chef's Hat

There are three phases per round (*Shift*):

1. Start of the Shift: cards and roles are dealt to the players. Players exchange up to 2 cards with each other depending on their role;
2. Making Pizzas: The bulk of the game. Players take turns in placing cards according to the rules on the board until all of them empty their hands;

Chef's Hat

There are three phases per round (*Shift*):

1. Start of the Shift: cards and roles are dealt to the players. Players exchange up to 2 cards with each other depending on their role;
2. Making Pizzas: The bulk of the game. Players take turns in placing cards according to the rules on the board until all of them empty their hands;
3. End of Shift: a cleanup step where points are scored and roles are reassigned. No players' agency in this phase.

Chef's Hat

There are three phases per round (*Shift*):

1. Start of the Shift: cards and roles are dealt to the players. Players exchange up to 2 cards with each other depending on their role;
2. Making Pizzas: The bulk of the game. Players take turns in placing cards according to the rules on the board until all of them empty their hands;
3. End of Shift: a cleanup step where points are scored and roles are reassigned. No players' agency in this phase.

We deal with the second phase only.

The Data

A total of 7819 unique triples of the form (*Player_Hand*, *Board_Before*, *Possible_Actions*) obtained by letting four agents which choose random moves play against each other for 100 matches.

The Data

A total of 7819 unique triples of the form (*Player_Hand*, *Board_Before*, *Possible_Actions*) obtained by letting four agents which choose random moves play against each other for 100 matches.

We only use the 221 triples from the last 3 matches as a test set.

The Data

A total of 7819 unique triples of the form (*Player_Hand*, *Board_Before*, *Possible_Actions*) obtained by letting four agents which choose random moves play against each other for 100 matches.

We only use the 221 triples from the last 3 matches as a test set.

The first two are lists of 17 and 11 natural numbers from 0 to 13. All elements represent the value of a card, except for 0 (missing card) and 12 (Joker card).

The Data

A total of 7819 unique triples of the form (*Player_Hand*, *Board_Before*, *Possible_Actions*) obtained by letting four agents which choose random moves play against each other for 100 matches.

We only use the 221 triples from the last 3 matches as a test set.

The first two are lists of 17 and 11 natural numbers from 0 to 13. All elements represent the value of a card, except for 0 (missing card) and 12 (Joker card).

Possible_Actions are lists of elements CX;QY;JZ and *pass*, where the former represents the move which consists in playing Y cards with value X and Z Joker cards.

The Data

A total of 7819 unique triples of the form (*Player_Hand*, *Board_Before*, *Possible_Actions*) obtained by letting four agents which choose random moves play against each other for 100 matches.

We only use the 221 triples from the last 3 matches as a test set.

The first two are lists of 17 and 11 natural numbers from 0 to 13. All elements represent the value of a card, except for 0 (missing card) and 12 (Joker card).

Possible_Actions are lists of elements CX;QY;JZ and *pass*, where the former represents the move which consists in playing Y cards with value X and Z Joker cards.

We feed prompts that are made of 3 different templates of increasing complexity with 5/10/15 pairs of game states and corresponding legal moves to Llama 3.1 8B Instruct.

Example

Player Hand: (0, 0, 5, 6, 7, 7, 7, 8, 9, 9, 9, 10, 10, 10, 11, 11, 11);

Board State: (13, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)

Legal moves: ['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']

Objectives

Objectives

It has been observed that choosing close examples that are semantically relevant to test instances via KNN (in the embedding space, either via euclidean distance or cosine similarity) can greatly improve the performance of ICL.

Objectives

It has been observed that choosing close examples that are semantically relevant to test instances via KNN (in the embedding space, either via euclidean distance or cosine similarity) can greatly improve the performance of ICL.

On the other hand, semantic diversity between demonstrations has also been shown to play a key role in ICL performances.
We propose graph-based approaches to demonstration selection with the goal of choosing examples that are:

Objectives

It has been observed that choosing close examples that are semantically relevant to test instances via KNN (in the embedding space, either via euclidean distance or cosine similarity) can greatly improve the performance of ICL.

On the other hand, semantic diversity between demonstrations has also been shown to play a key role in ICL performances.
We propose graph-based approaches to demonstration selection with the goal of choosing examples that are:

1. Relevant to a given query;

Objectives

It has been observed that choosing close examples that are semantically relevant to test instances via KNN (in the embedding space, either via euclidean distance or cosine similarity) can greatly improve the performance of ICL.

On the other hand, semantic diversity between demonstrations has also been shown to play a key role in ICL performances.
We propose graph-based approaches to demonstration selection with the goal of choosing examples that are:

1. Relevant to a given query;
2. Good summaries of abstract concepts of all the available examples;

Objectives

It has been observed that choosing close examples that are semantically relevant to test instances via KNN (in the embedding space, either via euclidean distance or cosine similarity) can greatly improve the performance of ICL.

On the other hand, semantic diversity between demonstrations has also been shown to play a key role in ICL performances.
We propose graph-based approaches to demonstration selection with the goal of choosing examples that are:

1. Relevant to a given query;
2. Good summaries of abstract concepts of all the available examples;
3. Diverse to each other.

Graph Theory Tools

Graph Theory Tools

We mainly use two tools from graph theory:

Graph Theory Tools

We mainly use two tools from graph theory:

1. The PageRank centrality measure: a variant of eigenvector centrality;

Graph Theory Tools

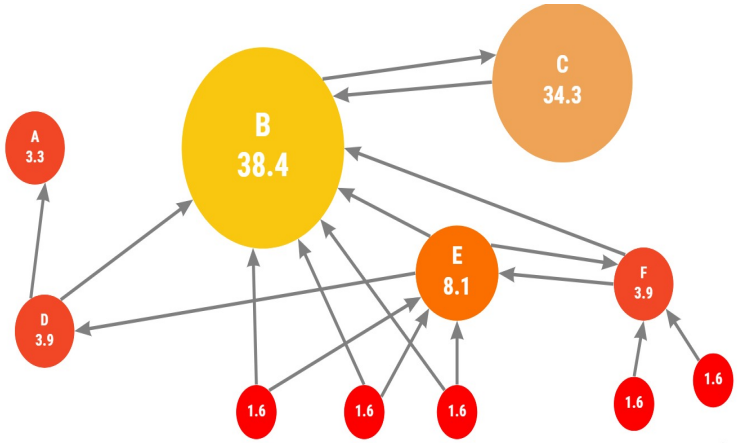
We mainly use two tools from graph theory:

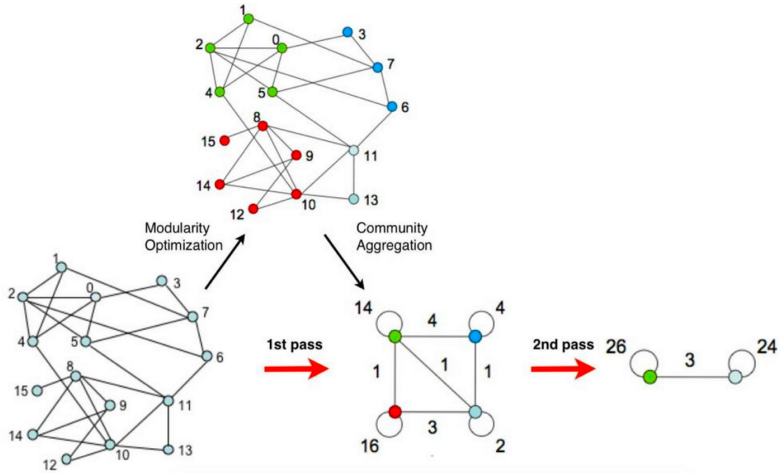
1. The PageRank centrality measure: a variant of eigenvector centrality;
2. The Louvain method for community detection: a fast heuristic method for partitioning graphs in communities with high modularity.

Graph Theory Tools

We mainly use two tools from graph theory:

1. The PageRank centrality measure: a variant of eigenvector centrality;
2. The Louvain method for community detection: a fast heuristic method for partitioning graphs in communities with high modularity.





Main approach

The proposed approach follows these steps:

1. Build the *Demonstration Graph* according to a *resolution* parameter R;

Main approach

The proposed approach follows these steps:

1. Build the *Demonstration Graph* according to a *resolution* parameter R;
2. For each test instance, extract a subgraph of the *Demonstration Graph* according to a *radius* parameter r;

Main approach

The proposed approach follows these steps:

1. Build the *Demonstration Graph* according to a *resolution* parameter R;
2. For each test instance, extract a subgraph of the *Demonstration Graph* according to a *radius* parameter r;
3. Return the top 5/10/15 demonstrations from the extracted subgraph according to the PageRank metric.

Main approach

The proposed approach follows these steps:

1. Build the *Demonstration Graph* according to a *resolution* parameter R;
2. For each test instance, extract a subgraph of the *Demonstration Graph* according to a *radius* parameter r;
3. Return the top 5/10/15 demonstrations from the extracted subgraph according to the PageRank metric.

Observation

If links between nodes represent similarity, the nodes with the highest PageRank scores are the most similar to nodes that are similar to many nodes.

Main approach

The proposed approach follows these steps:

1. Build the *Demonstration Graph* according to a *resolution* parameter R;
2. For each test instance, extract a subgraph of the *Demonstration Graph* according to a *radius* parameter r;
3. Return the top 5/10/15 demonstrations from the extracted subgraph according to the PageRank metric.

Observation

If links between nodes represent similarity, the nodes with the highest PageRank scores are the most similar to nodes that are similar to many nodes.

We use the Hamming distance between concatenations of Board states and Players' hands.

Variants

Variants

1. Add weights to edges (i,j) monotonically decreasing with respect to $d(i,j)$;

Variants

1. Add weights to edges (i,j) monotonically decreasing with respect to $d(i,j)$;
2. Use the Louvain method on the extracted subgraph and return the top nodes (PageRank-wise) from the largest detected communities.

Variants

1. Add weights to edges (i,j) monotonically decreasing with respect to $d(i,j)$;
2. Use the Louvain method on the extracted subgraph and return the top nodes (PageRank-wise) from the largest detected communities.

Observation

The Louvain variant aims at increasing the diversity between prompted demonstrations.

Results

	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: IoU scores of the two baselines.

Results

	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: 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.

Results

		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: 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.

A candidate explanation

The results show that in this case closeness to the test instances is the most impactful factor in performances, while other parameters and approaches produce marginal differences.

A candidate explanation

The results show that in this case closeness to the test instances is the most impactful factor in performances, while other parameters and approaches produce marginal differences.

This is probably due to a poor choice of distance: game states with a Hamming distance of 1 can have vastly different sequences of legal moves.

A candidate explanation

The results show that in this case closeness to the test instances is the most impactful factor in performances, while other parameters and approaches produce marginal differences.

This is probably due to a poor choice of distance: game states with a Hamming distance of 1 can have vastly different sequences of legal moves.

This effect quickly becomes larger and larger as the Hamming distance increases.

Further developments

A number of topics can be further investigated:

Further developments

A number of topics can be further investigated:

- Different choice of metric;

Further developments

A number of topics can be further investigated:

- Different choice of metric;
- Different choice of node scoring;

Further developments

A number of topics can be further investigated:

- Different choice of metric;
- Different choice of node scoring;
- Systematic ways to set parameters R and r;

Further developments

A number of topics can be further investigated:

- Different choice of metric;
- Different choice of node scoring;
- Systematic ways to set parameters R and r;
- For weighted approaches, a different degradation method of the weights;

Further developments

A number of topics can be further investigated:

- Different choice of metric;
- Different choice of node scoring;
- Systematic ways to set parameters R and r;
- For weighted approaches, a different degradation method of the weights;
- For the Louvain method, a systematic way to choose the resolution parameter γ;

Further developments

A number of topics can be further investigated:

- Different choice of metric;
- Different choice of node scoring;
- Systematic ways to set parameters R and r;
- For weighted approaches, a different degradation method of the weights;
- For the Louvain method, a systematic way to choose the resolution parameter γ;
- Alternatives to the Louvain method for (potentially overlapping) community detection;

Further developments

A number of topics can be further investigated:

- Different choice of metric;
- Different choice of node scoring;
- Systematic ways to set parameters R and r;
- For weighted approaches, a different degradation method of the weights;
- For the Louvain method, a systematic way to choose the resolution parameter γ;
- Alternatives to the Louvain method for (potentially overlapping) community detection;
- Use of different LLMs.

Further developments

A number of topics can be further investigated:

- Different choice of metric;
- Different choice of node scoring;
- Systematic ways to set parameters R and r;
- For weighted approaches, a different degradation method of the weights;
- For the Louvain method, a systematic way to choose the resolution parameter γ;
- Alternatives to the Louvain method for (potentially overlapping) community detection;
- Use of different LLMs.
- Application of the methods to different, more popular, domains.

Thank you for your attention!