Code Explanation Assessment Using LLMs and Analogical Reasoning on OIE Graphs

Nisrine Ait Khayi[1,*,†], Vasile Rus[1,†]

[1]The University of Memphis, 375 Dunn Hall, Memphis, TN 38152, USA

Abstract

Assessing self-explanations of code is a critical task in educational Natural Language Processing (NLP), essential for providing automated feedback in programming education. While recent studies demonstrate that LLMs exhibit analogical reasoning capabilities, their application to educational code explanation assessment remains underexplored. To address this gap, we propose a novel three-stage approach: (1) prompting the LLM to extract Open Information Extraction (OIE) units from student and expert explanations, (2) prompting the LLM to construct semantic graphs from these units, and (3) employing LLM-based analogical reasoning to assess explanation similarity. We evaluate our approach on the Self-code corpus using Pearson and Spearman correlations. Our method achieves correlations of 0.8 and 0.76, respectively, significantly outperforming supervised models like BERT (0.74/0.75) and unsupervised approaches like text-embedding-ada-002 (0.67/0.64). These results demonstrate the effectiveness of structured semantic representation combined with analogical reasoning for educational assessment tasks.

Keywords

analogical reasoning, OIE graphs, automated assessment, large language models, code comprehension

1. Introduction

Assessing students' self-explanations of learning content is a critical task in educational Natural Language Processing (NLP) and education technologies that prompt students for natural language input. Indeed, accurate assessment of learner's self-explanations is critical in highly-adaptive education technologies such as Intelligent Tutoring Systems (ITSs) that prompt learners for such explanations. Accurate assessment enables these ITSs to provide tailored feedback and hints to each individual student, which is needed to maximize student learning outcomes[1, 2, 3].

One approach to automatically assessing student self-explanations consists of evaluating the semantic equivalence between, for instance, two explanations of a code statement: a student explanation versus a benchmark or gold explanation, e.g., generated by an expert. That is, a student self-explanation is assessed against a benchmark explanation whose correctness value is known. Specifically, a semantic similarity score is computed, for instance, on a scale from 0 (no semantic similarity) to 5 (complete semantic similarity) [4]. Other scoring scales have been used, such as a normalized continuous score between 0 and 1 (1=semantic equivalence). It should be noted that this semantic similarity approach is the most widely used approach for assessing free-text self-explanations.

Many approaches have been explored to address the semantic similarity task over the past several decades[5, 6, 7].Recent advancements in foundation models have leveraged fine-tuned pre-trained language models to capture the similarity on a sentence level, yielding state-of-the-art performance. For instance, Lee and colleagues [8] introduced a contrastive learning framework that optimizes the weighted sum of contextualized token distances, enhancing both similarity scores and interpretability. Rep and colleagues [9] improved performance by applying a novel Truncated Model Fine-Tuning (TMFT) method for ELECTRA [10]. The TMFT method uses mean pooling over the last layer's embedding.TMFT improves the Spearman correlations by over 8 points while increasing parameter efficiency on the

AISEER'25: 2nd International Workshop on AI in Society, Education and Educational Research, Bologna, Italy

*Corresponding author

†These authors contributed equally.

ntkhynyn@memphis.edu (N. A. Khayi); vrus@memphis.edu (V. Rus)
0000-0002-7809-7178 (N. A. Khayi); 0009-0002-4739-0440 (V. Rus)

STS benchmark. Despite these successes, Gatto et al.[11] showed that generative LLMs outperform existing encoder-based models, revealing limitations of these encoder-based approaches in their learning capabilities. These findings suggest that while LLMs excel at pattern recognition, they may benefit from incorporating more structured reasoning mechanisms, such as analogical reasoning, which humans naturally use to understand and compare complex conceptual relationships.

Analogical reasoning, a cornerstone of human cognitive intelligence, enables knowledge adaptation [12], decision-making [13], and problem-solving [14]. According to the Structure Mapping Theory (SMT) [15], humans conduct analogical reasoning using the shared relational structure between two domains. Mental Models Theory [16] focuses on how people create mental models of the world to reason about structures and functions. This theory supports functional analogical reasoning by suggesting that humans can simulate mental models of complex dynamic systems based on how they function in real life. For example, the function of the 'heart pumping blood' is analogous to the function of a 'pump pumping water' despite differences in their structures. The multi-constraint theory [17] introduces the concept of constraints satisfaction, where the analogical reasoning is guided by: (1) direct similarity between the elements involved, (2) the structural parallels in the roles between the source and target domains, and (3) the purpose constraint implying that analogical thinking is guided by the goal of the reasoner.

Recent studies have shown that LLMs exhibit analogical reasoning capabilities, particularly through in-context learning [18]. For example, Yasunaga et al. [14] demonstrated that analogical prompting instructing LLMs to self-generate relevant exemplars before solving the problem using in-context learning, improves their reasoning process. Qin et al. [19] demonstrated that self-generated random examples can achieve comparable or even better performance. However, investigating the LLMs' potential to perform analogical reasoning in semantic similarity contexts remains underexplored for the following reasons: (1) the existing LLMs work on semantic similarity focuses on surface-level similarity, and (2) the need for more robust evaluation to validate the LLMs' analogical reasoning capabilities. To address this gap, we propose a novel approach based on the Structure Mapping and Mental Model theories and which combines analogical reasoning with LLMs-based prompting to enhance the assessment of code self-explanations. More specifically, our approach uses LLMs to infer structured representations of code explanations in the form of Open Information Extraction (OIE) graphs, which use subject-predicate-object triplets as their basic unit structure. The proposed approach differs from existing LLM-based prompting approaches by extracting OIE units from self-explanations. Constructing OIE graphs for structured representations enables more interpretable similarity assessment and also allows us to make use of the analogical reasoning of the LLMs on these structured representations.

Our main contributions are as follows:

- Explore the potential of LLMs to extract OIE triplets to represent the semantics of self-explanations as structured components.
- Explore the potential of LLMs to construct OIE graphs from extracted triplets that capture relational and semantic roles reflecting relationships within self-explanations.
- Explore the potential of LLMs to conduct functional and structural analogical reasoning for assessment purposes of explanations.

The remainder of the paper is organized as follows: Section 2 reviews related works, Section 3 describes the proposed method, Section 4 presents the experiments and results, and Section 5 concludes this work with future directions.

2. Related Works

Large Language Models (LLMs) have recently gained prominence in education, particularly in the automated assessment task. For instance, Xie et al.[20]proposed an LLM-based grading system based on three stages: (1) rubric generation, where students' answers are incorporated into the rubric design; (2) grading, where the generated rubric guides the LLM to grade students' answers using various prompting strategies (one shot-prompts, self-reflection prompts where the LLM is asked to reflect on the previous output, batching prompts where students answers are processed in batches and graded together by the LLM); and (3) post-grading review, where the LLM defines the unreasonable results and sends them back to re-grading. Their empirical experiments demonstrated the effectiveness of their approach, providing new insights for the assessment task using LLMs. Similarly, Cohn et al. [21]employed Chain-of-Thought (CoT) prompting and an active learning approach for grading and explaining the assessment of responses in science. The model's reasoning was used to elicit explanations for its responses, enabling teachers to provide feedback to students. Xiao et al.[22] showcased the effectiveness of LLMs in the Automated Essay Scoring (AES) task, emphasizing the importance of providing comprehensive contexts, clear rubrics, and high-quality examples. Their research highlighted the capability of LLMs to provide explanations and feedback to students besides the essay's score. Additionally, Katuka et al. [23] explored the potential of quantized LLMs in improving the performance of the automatic grading and feedback generation downstream tasks. They fine-tuned 4-bit quantized Llama-2 for automatic grading and feedback generation, using supervised instruction fine-tuning. Their empirical results showed a reduction in the error rates in grading by 3% and their approach outperforming traditional LLMs like GPT-2 in generating feedback with improvements of 0.19,0.2, and 0.12 in BLEU, ROUGE-1, and ROUGE-2 scores respectively, for the SAR dataset. For the Proprietary dataset, they achieved improvements of 0.64, 0.46, and 0.65 in BLEU, ROUGE-1, and ROUGE-2 scores respectively. Oli et al.[24] used LLMs to compare students' self-explanations and experts' self-explanations in the context of line by-line explanation of computer programs. Their results demonstrated the superior performance of the LLMs, when prompted in few-shot and Chain of Thoughts settings, over the fine-tuned encoder-based models.

3. Approach

Our approach relies on LLMs at different phases of its process. First, we prompt the LLM to extract Open Information Extraction (OIE) triplets from the student and expert explanations, capturing their semantics. Second, we prompt the LLM to construct OIE graphs, reflecting relationships in the explanations. Third, we prompt the LLM to perform structural and functional analogical reasoning on OIE graphs to assess code self-explanations.

3.1. The Open Information Extraction Phase

Open Information Extraction (OIE) systems map a sentence S= $x_1, ..., x_n$ with a relational triplet (arg_1, *rel*, arg_2), where arg_1 and arg_2 are noun phrases created from tokens x_i in S, and rel represents the semantic relationship between arg_1 and arg_2 [25]. In this phase, we prompt the LLM to extract OIE triplets from both student and expert self-explanations. The OIE prompt was refined through an iterative refinement process to optimize its effectiveness. We started with a simple prompt, which didn't yield very good results. Providing some examples, i.e., few-shot prompting where the model is provided with few examples to facilitate the understanding of the task [26], helped. Additionally, we incorporated a specialized role within the prompting framework [27], and we systematically varied instruction phrasing and example selections to optimize triplet extraction quality.

3.2. The OIE Graph Construction Phase

In this phase, the LLM is prompted to construct OIE graphs for both student and expert explanations, representing the key relationships and interactions within the explanations.

Using OIE triplets in the form of (arg_1, *rel*, arg_2), we instruct the LLM to identify the most significant OIE triplets by prompting it to extract those representing core conceptual relationships and main operations in the explanations. These triplets are then structured into nodes, representing arguments (arg_1 and arg_2), and edges, representing their semantic relationships (*rel*). Additionally, we incorporated a few examples in a JSON format to facilitate the LLM's understanding of the OIE graphs' construction.

3.3. The Analogical Reasoning Phase

Functional analogical reasoning looks for analogous functions and roles that nodes or edges play in the OIE student and expert graphs. It assesses whether the explanations describe the same purpose and behavior of the code, even if their phrasing or level of detail differs. For example:

- **Code statement**: count += 1.
- **User explanation**: The variable count is incremented by one each time the loop runs.
- **Expert explanation**: Each loop iteration increases count by one.
- **OIE graph generation**: Using Open Information Extraction, key entities (nodes) and relationships (edges) are automatically extracted from each explanation.
- **OIE graph1**:
  - Node1: ”count”, Node2: ”one”, Edge: ”incremented by”.
  - Node1: ”loop”, Node2: ”each time”, Edge: ”runs”.
- **OIE graph2**:
  - Node1: ”count”, Node2: ”one”, Edge:”increases by”
  - Node1: ”loop iteration”, Node2:”count”, Edge:”increases”

These graphs are functionally analogous, both capturing the essential relationship between loop execution and count incrementing.

Structural analogical reasoning compares the organizational structure between two OIE graphs, such as the arrangement of nodes and edges, even if they describe different code functions.

The functional and structural analogical reasoning prompt was refined through an iterative process, where we systematically varied instruction phrasing, role definitions, and example selections, evaluating each version’s ability to perform accurate analogical reasoning.

3.4. Methodology Illustration

To illustrate our complete methodology, consider the following example :

- **Code statement**: user_info[’name’] = input (”Enter your name: ”).
- **Student explanation**: The user_info dictionary stores data entered by the user
- **Expert explanation**:The name field in user_info dictionary is updated with input provided by the user.
- **Student Explanation OIE triplets is shown in Listing 1:**
  - (user_info, stores, data)
  - (data, entered by, user)
- **Expert Explanation OIE triplets is shown in Listing 2**:
  - (name field, in, user_info dictionary)
  - (user_info, updated with, input)
  - (input, provided by, user)
- **OIE student explanation graph**:

Listing 1: JSON Student Graph Representation

```
{
  "nodes": [
    {"id": "user_info", "label": "dictionary"},
    {"id": "data", "label": "value"},
    {"id": "user", "label": "source"}
  ],
  "edges": [
    {"source": "user_info", "target": "data", "relation": "stores"},
    {"source": "data", "target": "user", "relation": "entered by"}
  ]
}
```

- **OIE expert explanation graph**:

Listing 2: JSON Expert Graph Representation

```
{
  "nodes": [
    {"id": "name field", "label": "dictionary field"},
    {"id": "user_info", "label": "dictionary"},
    {"id": "input", "label": "value"},
    {"id": "user", "label": "source"}
  ],
  "edges": [
    {"source": "user_info", "target": "input", "relation": "updated with"},
    {"source": "input", "target": "user", "relation": "provided by"}
    {"source": "name field", "target": "user info dictionary", "relation": "in"}
  ]
}
```

- **Similarity Score:** The graphs show different focuses - the student emphasizes data storage while the expert emphasizes field updating. These represent different functional aspects of the same code. Structural patterns differ in node relationships. Similarity score :0.45. The OIE triplets, OIE graphs, and the semantic score are generated by the LLM.

4. Experiments and Results

To evaluate the performance of our proposed method, we have conducted several experiments with the Self-code corpus.

4.1. Dataset

The Self-code corpus[28] consists of sentence pairs of student and expert self-explanations of Java code examples, along with semantic similarity judgments using a 1-5 scale, where 1 means the student and expert self-explanations are not similar, whereas 5 means they are semantically equivalent. The expert explanations are from a collection of expert-annotated examples from a catalog of interactive learning content[29]. A crowdsourcing approach has been followed to collect explanations from students with different backgrounds and skills. Examples of student and expert explanations of code lines are shown in Table 1.

Table 1
Examples of student and expert explanations of code lines

Example code line	Expert explanation	Student explanation
Point1 point = new Point1()	The variable point holds a reference to a Point1 object.	Create a new Point1.
System.out.println (” The integer is positive”)	This statement prints that the integer is positive.	Print that the number is positive if it is greater than 0.
String fullname =” John Smith”	We define a string variable to hold the name.	Sets the fullName string to John Smith.
translate (11,6)	This line invokes the method translate of the point.	moves the X, and Y points by adding (11,6) point.

4.2. Experimental Setup

LLM-based prompting: We evaluated our proposed analogical reasoning approach against Chain-of-Thought (CoT) prompting in a few-shot setting using 3 examples per prompt. We randomly selected 100 instances from the Self-code dataset to balance analytical depth with practical constraints of cost and computational time. The selected sample's distribution across similarity scores is similar to the original distribution. We experimented with multiple state-of-the-art LLMs: GPT-4o, GPT-4-turbo [30], mistral-large-latest[31], and llama-3.1-70b [32]. For the GPT models, temperature was set to 0 to ensure deterministic outputs. Default temperature settings were used for Mistral and LLaMA models. The CoT baseline prompting approach asks the LLMs to compute the similarity scores between student and expert self-explanations step by step, using the phrase "Let's think step by step".

Supervised STS approaches: We compare our analogical prompting-based approach to: (1) ALBERT[33], (2) RoBERTa[34], (3) BERT [35], and (4) CodeBERT[36]. We used the base versions of the pretrained models that are fine-tuned for 3 epochs. The AdamW optimizer with a learning rate of 2e-5 has been used. The self-code dataset contains 1770 instances. The models were tested using the same test data for the analogical reasoning prompting-based approach. The remaining dataset (1670 'instances) has been used for training (80%) and validation (20%).

Unsupervised STS approaches: Student and expert self-explanations are encoded using state-of-the-art models on the STS task and then compared using the cosine similarity. We experimented with: (1) Open AI embeddings (openai-ada-002 and openai-embedding-3-large), (2) the pretrained all-mpnet-base-v2 finetuned on a 1B sentence pair dataset, and (3) SimCSE[5], which leverages enhanced sentence representations using contrastive learning. We used SimCSE-BERT and SimCSE-Roberta.

All approaches produce similarity scores on the same 100-instance test dataset. LLM outputs are normalized to the 0-1 range. We report Spearman's rank correlation and Pearson's correlation between the predicted scores and ground-truth similarity scores (normalized from 1-5 scale to 0-1 scale), in the dataset's subset.

4.3. Experimental Results

Table 2 presents the comprehensive evaluation results comparing our proposed OIE-analogical reasoning approach against Chain-of-Thought (CoT) prompting and established baseline methods across multiple model categories.

LLM-based Approaches
According to Table 2, our analogical reasoning approach on OIE graphs consistently outperforms the CoT baseline across most LLMs. GPT-4o achieves the strongest performance with Pearson correlation of 0.8 and Spearman correlation of 0.76, representing improvements of 8.1% and 8.6% respectively over CoT. LLaMA shows substantial improvements of 10.8% in both metrics (0.72 vs 0.65). Notably, GPT-4-Turbo shows mixed results, with slightly lower Pearson correlation (0.72 vs 0.73) but improved Spearman correlation (0.7 vs 0.66). This suggests that our approach may be sensitive to some LLMs' architectures and training procedures. Mistral demonstrates the most modest improvements, achieving 4.9% and 3.3% gains in Pearson and Spearman correlations, respectively, suggesting limitations in analogical reasoning and information extraction capabilities for this task. The findings also underscore the superior analogical reasoning ability, and the information extraction of GPT-4 models compared to the open-source models: Mistral and LLaMA. These results align with prior findings on the analogical reasoning performance of GPT-4 models[37].

Comparison with Supervised and Unsupervised Baselines
As evidenced by Table 2, our approach significantly outperforms both supervised and unsupervised baselines. Compared to the best-performing supervised model (BERT: 0.74, 0.75), GPT-4o with our

Table 2
Results comparing our OIE-analogical reasoning approach with CoT approach, fine-tuned supervised models and unsupervised models

Models	Pearson	Spearman
GPT-4o Experiments		
OIE-analogical reasoning	**0.8**	**0.76**
CoT	0.74	0.7
GPT-4-Turbo Experiments		
OIE-analogical reasoning	0.72	0.7
CoT	0.73	0.66
Mistral Experiments		
OIE-analogical reasoning	0.64	0.62
CoT	0.61	0.6
LLaMA Experiments		
OIE-analogical reasoning	0.72	0.72
CoT	0.65	0.65
Supervised STS Experiments		
ALBERT	0.67	0.72
RoBERTa	0.69	0.69
BERT	0.74	0.75
CodeBERT	0.71	0.71
Unsupervised STS Experiments		
OpenAI-ada-002	0.66	0.64
OpenAI-embedding-3-large	0.67	0.64
all-mpnet-base-v2	0.51	0.50
SimCSE-BERT	0.52	0.50
SimCSE-ROBERTA	0.57	0.55

approach achieves improvements of 8.1% and 1.3% in Pearson and Spearman correlations. Against other supervised models, the improvements are even more substantial: 19.4% and 5.6% over ALBERT, 15.9% and 10.1% over RoBERTa, and 12.7% and 7.0% over CodeBERT. Unsupervised approaches perform considerably worse, with the best OpenAI embedding (embedding-3-large) achieving only 0.67 and 0.64 correlations, representing a performance gap of 19.4% and 18.8% compared to our GPT-4o implementation. The superior performance of the LLMs can be explained by several factors: (1) the limited size of the training dataset of the fine-tuned supervised models, which may be insufficient for capturing semantic relationships in the programming self-explanations, (2) LLMs leverage extensive pretraining and complex reasoning capabilities, (3) our structured OIE-graph approach captures more nuanced semantic understanding in comparison to the surface-level cosine similarity in unsupervised models.

OIE Graphs Ablation

Table 3 demonstrates the critical role of OIE graph construction in our approach. When performing analogical reasoning directly on self-explanations without OIE graphs, all models show performance degradation. GPT-4o experiences drops of 5.0% and 2.6% in Pearson and Spearman correlations, respectively, while LLaMA shows more substantial decreases of 5.6% and 9.7%. These results confirm that structured knowledge representation through OIE graphs is essential for effective analogical reasoning in this domain.

Table 3
Experimental results of analogical reasoning on self-explanations without using OIE graphs

Model	Pearson	Spearman
GPT-4o	0.76	0.74
GPT-4-Turbo	0.5	0.53
Mistral	0.6	0.54
LLaMA	0.68	0.65

4.4. Error Analysis

We analyzed failed predictions with low confidence scores (<= 0.6) to understand where LLMs fail in performing accurate similarity assessment.

Confidence Computation
For any student self-explanation and expert self-explanation pair, we prompt the LLM to generate a verbalized confidence score.

Failure Analysis
We analyzed 24 low-confidence predictions (out of 100 instances) from GPT-4o to understand where our proposed approach fails in performing an accurate similarity assessment. The analysis revealed three error types: OIE extraction errors (33.3%), structural analogical reasoning errors (29.2%), and functional analogical reasoning errors (37.5%). Functional reasoning errors were more prevalent, indicating the LLMs' challenges in recognizing semantic equivalence between different expressions of the same concept. For example, the model may fail to map ”loop processes numbers” to ”iteration over values,” despite both describing the same algorithmic behavior. OIE extraction errors include incorrect or incomplete triplet extraction from explanations. For instance, extracting '(variable, set to, 0)' when the explanation describes initialization with array values. While less frequent, these errors had a significant impact on performance. Structural analogical errors involve difficulty in aligning graph structures, even when the extracted triplets are correct. The model may struggle to identify corresponding nodes and edges between student and expert graphs with different organizational patterns, leading to incorrect similarity assessment.

5. Conclusion

We proposed a novel approach that combines analogical reasoning with Open Information Extraction (OIE) graphs to enhance LLM-based code self-explanation assessment. Experimental evaluation on the Self-code corpus demonstrates significant improvements: 8.1% and 8.6% gains over Chain-of-Thought prompting using GPT-4o, and up to 19.4% improvement over both supervised and unsupervised baselines, highlighting the effectiveness of structured analogical reasoning for this task. The ablation study confirms that OIE graph construction is essential, with performance drops of 5.0% when removed from the pipeline. Error analysis of low-confidence failed predictions identified three categories of errors: functional analogical reasoning errors (37.5%), OIE extraction errors (33.3%), and structural analogical reasoning errors (29.2%). These findings suggest that while our approach is effective, there remain challenges in semantic mapping and robust triplet extraction for programming domain explanations. Building on these findings, we plan to: (1) explore LLMs' generated explanations to provide tailored hints and feedback for students, aiming to improve their learning outcomes, (2) implement advanced prompt optimization strategies to enhance the LLMs' capabilities in open information extraction, (3) enhance the LLM prompting for analogical reasoning, and (4) add richer visualizations to convey the results more efficiently.

This work contributes to the AI-education field by demonstrating how structured reasoning approaches can enhance LLM capabilities for programming-based assessment tasks.

Generative AI Statement

The authors acknowledge the use of generative artificial intelligence (AI) tools during the preparation of this research paper. Claude was used to rephrase selected paragraphs, polish sentences, and generate illustrative percentages presented in the results section. In addition, the authors discussed the approach structure with the AI tool, which proposed to include an example that was subsequently adapted and verified by the authors. All research ideas, conceptual development, data analysis, interpretation of findings, and final writing were carried out and validated solely by the authors, who take full responsibility for the accuracy and integrity of the paper.

References

[1] D. Boud, E. Molloy (Eds.), Feedback in Higher and Professional Education, Routledge, 2012.

[2] V. J. Shute, Focus on formative feedback, Review of educational research 78 (2008) 153–189.

[3] I. Azaiz, N. Kiesler, S. Strickroth, Feedback-generation for programming exercises with gpt-4, in: Proceedings of the 2024 on Innovation and Technology in Computer Science Education V, 2024, pp. 31–37.

[4] E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez-Agirre, R. Mihalcea, G. Rigau, J. Wiebe, Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation, in: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), , San Diego, California, Association for Computational Linguistics, 2016, pp. 497–511.

[5] T. Gao, X. Yao, D. Chen, Simcse: Simple contrastive learning of sentence embeddings, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, , Online and Punta Cana, Dominican Republic, Association for Computational Linguistics, 2021, pp. 6894–6910.

[6] Y. Chen, Y. Zhang, B. Wang, Z. Liu, H. Li, Generate, discriminate and contrast: A semi-supervised sentence representation learning framework, in: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, , Abu Dhabi, United Arab Emirates, Association for Computational Linguistics, 2022, pp. 8150–8161.

[7] M. Mohebbi, S. N. Razavi, M. A. Balafar, Computing semantic similarity of texts based on deep graph learning with ability to use semantic role label information, Scientific Reports 12 (2022) 1–11.

[8] S. Lee, D. Lee, S. Jang, H. Yu, Toward interpretable semantic textual similarity via optimal transport-based contrastive sentence learning, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1, Association for Computational Linguistics, 2022, pp. 5969–5979.

[9] I. Rep, D. Dukić, J. Šnajder, Are electra's sentence embeddings beyond repair? the case of semantic textual similarity, Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA. Association for Computational Linguistics (2024) 9159–9169.

[10] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, Electra: pretraining text encoders as discriminators rather than generators, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.

[11] J. Gatto, O. Sharif, P. Seegmiller, P. Bohlman, S. Preum, Text encoders lack knowledge: Leveraging generative llms for domain-specific semantic textual similarity, in: Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), Singapore, Association for Computational Linguistics, 2023, pp. 277–288.

[12] D. R. Hofstadter, Analogy as the core of cognition, The analogical mind: Perspectives from cognitive science (2001) 499–538.

[13] P. Hansen-Estruch, A. Zhang, A. Nair, P. Yin, S. Levine, Bisimulation makes analogies in goal-conditioned reinforcement learning, in: International Conference on Machine Learning, PMLR, 2022, pp. 8407–8426.

[14] M. Yasunaga, X. Chen, Y. Li, P. Pasupat, J. Leskovec, P. Liang, E. H. Chi, D. Zhou, Large language models as analogical reasoners, arXiv preprint arXiv:2310.01714 (2023).

[15] D. Gentner, Structure-mapping: A theoretical framework for analogy, Cognitive science 7 (1983) 155–170.

[16] P. N. Johnson-Laird, Mental models, Cambridge University Press, Cambridge, UK, 1983.

[17] K. J. Holyoak, P. Thagard, The analogical mind: Perspective and scope, Psychological Review 102 (1995) 428–444.

[18] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, others, D. Amodei, Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[19] C. Qin, W. Xia, T. Wang, F. Jiao, Y. Hu, B. Ding, R. Chen, S. R. Joty, Relevant or random: Can llms truly perform analogical reasoning?, 2024. arXiv:abs/2404.12728.

[20] W. Xie, J. Niu, C. J. Xue, N. Guan, Grade like a human: Rethinking automated assessment with large language models, 2024. arXiv:2405.19694, arXiv preprint.

[21] C. Cohn, N. Hutchins, T. Le, G. Biswas, A chain-of-thought prompting approach with llms for evaluating students' formative assessment responses in science, Proceedings of the AAAI Conference on Artificial Intelligence 38 (2024) 23182–23190.

[22] C. Xiao, W. Ma, S. X. Xu, K. Zhang, Y. Wang, Q. Fu, From automation to augmentation: Large language models elevating essay scoring landscape, 2024. arXiv:2401.06431, arXiv preprint (2024).

[23] G. A. Katuka, A. Gain, . Y. Y. Yu, Investigating automatic scoring and feedback using large language models, 2024. arXiv:2405.00602, arXiv preprint.

[24] P. Oli, R. Banjade, J. Chapagain, V. Rus, Proceedings of the 2024 aaai conference on artificial intelligence, PMLR 257 (2024) 118–128.

[25] P. Gamallo, An overview of open information extraction (invited talk), in: 3rd Symposium on Languages, Applications and Technologies, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2014.

[26] Y. Wang, Q. Yao, J. T. Kwok, L. M. Ni, Generalizing from a few examples: A survey on few-shot learning, ACM computing surveys (csur) 53 (2020) 1–34.

[27] B. Cabral, D. Claro, M. Souza, Exploring open information extraction for portuguese using large language models, in: Proceedings of the 16th International Conference on Computational Processing of Portuguese - Vol. 1, Santiago de Compostela, Galicia/Spain, Association for Computational Linguistics, 2024, pp. 127–136.

[28] J. Chapagain, Z. Risha, R. Banjade, P. Oli, L. Tamang, P. Brusilovsky, V. RUs, Selfcode: An annotated corpus and a model for automated assessment of self-explanation during source code comprehension, The International FLAIRS Conference Proceedings 36 (2023).

[29] A. Hicks, K. Akhuseyinoglu, C. Shaffer, P. Brusilovsky, Live catalog of smart learning objects for computer science education, in: Sixth SPLICE Workshop, 2020.

[30] OpenAI, Learning to reason with llms, 2024.

[31] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. D. L. Casas, others, W. E. Sayed, Mistral 7b, 2023. arXiv:2310.06825, arXiv preprint.

[32] H. Touvron, et al., Llama: Open and efficient foundation language models, 2023. arXiv:2302.13971, arXiv preprint.

[33] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut:ALBERT, A Lite BERT for Self-supervised Learning of Language Representations, ICLR, 2020.

[34] L. Zhuang, L. Wayne, S. Ya, Z. Jun, A robustly optimized bert pre-training approach with post-training, in: Proceedings of the 20th Chinese National Conference on Computational Linguistics, Huhhot, China, Chinese Information Processing Society of China, 2021, pp. 1218–1227.

[35] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), 2019, pp. 4171–4186.

[36] Z. Feng, D. Guo, D. Tang, N. Duan, X. Feng, M. Gong, L. Shou, B. Qin, T. Liu, D. Jiang, M. Zhou, Codebert: A pre-trained model for programming and natural languages, Findings of the Association for Computational Linguistics: EMNLP 2020, , Online. Association for Computational Linguistic (2020) 1536–1547.

[37] Z. Sourati, F. Ilievski, P. Sommerauer, Y. Jiang, Arn: Analogical reasoning on narratives, Transactions of the Association for Computational Linguistics 12 (2024) 1063–1086.