Online Knowledge Graph Construction and Visualization from Classroom Discourse with LLMs

Woo-Hyun Choi*, Jia Hong*, Kyung Kim*†, and Seunghyun Yoon*†
*Korea Institute of Energy Technology (KENTECH), Republic of Korea
{woohyunchoi, hja1617, kkim, syoon}@kentech.ac.kr
†No One Behind, Co., Ltd., Republic of Korea
{kkim, syoon}@nonebehind.com

***Abstract*—Classroom discourse provides rich evidence of how learners articulate and connect concepts, yet most discourse analysis remains offline due to the cost of transcription and manual coding. This paper addresses the practical systems problem of maintaining an interpretable representation of conceptual structure from a stream of instructional utterances under online constraints. We present a prototype pipeline that constructs an evolving entity–relation knowledge graph by using large language models for utterance normalization and schema-constrained extraction of concept entities and directed relations. To control redundancy under repeated updates, the pipeline incrementally consolidates semantically equivalent entities via embedding-based candidate generation with optional context-aware verification, and filters low-information relations to retain readable structure. We evaluate feasibility in a reproducible replay-based setting and report online measurements of final graph size, end-to-end update latency, and LLM call counts. We further study component contributions through an ablation suite aggregated over three random seeds.**

***Index Terms*—Classroom discourse, streaming analytics, knowledge graphs, large language models, entity and relation extraction.**

I. Introduction

Instructional talk in classrooms and tutoring sessions reveals how learners frame claims, introduce concepts, and revise explanations during interaction. Discourse-centric learning analytics and discourse analytics therefore treat language as a primary trace of learning processes [1], [2]. For instructors, however, discourse only becomes actionable when it can be summarized into a representation that remains legible within a class period. This need is consistent with the teacher-facing analytics and classroom orchestration literature, which emphasizes timely, compact views for inspection and reflection during ongoing activities [3]–[5].

In practice, classroom discourse analysis is still predominantly offline. Manual transcription and qualitative coding are costly, and results rarely feed back into instruction while interaction is unfolding [6], [7]. Although educational text mining has improved scalability, many pipelines assume post-hoc processing or curated written artifacts [8]. Streaming instructional discourse introduces additional systems constraints: utterances are often ill-formed, content evolves over time, and naively accumulating extracted concepts quickly yields redundant and cluttered outputs. Consequently, the core challenge is not only extracting concepts, but also *maintaining* an interpretable structure under repeated updates.

Large language models (LLMs) provide a practical mechanism for normalizing noisy language and extracting structured entities and relations with limited task-specific training [9]–[11]. In educational contexts, LLMs have shown promise for classroom dialogue analysis, while reliability and evaluation remain central concerns [12]. In an online setting, direct LLM extraction can still overwhelm users: overlapping mentions, near-synonyms, and generic relations accumulate rapidly unless explicit redundancy control is built into the update loop.

This paper presents a prototype system for streaming knowledge graph construction and visualization from classroom discourse. The system processes utterances in update batches, performs LLM-based normalization and schema-constrained extraction, and maintains a compact evolving graph through incremental entity consolidation and relation filtering. The design targets general instructional discourse settings (e.g., tutoring, group discussion, seminar interaction) where utterances arrive continuously and compactness is required for inspection. To enable controlled observation of online behavior, we evaluate feasibility using a replayable discourse stream and report online measurements of graph size outcomes, end-to-end update latency, and cost proxies.

Contributions. This paper makes the following contributions:

- A streaming pipeline that applies LLM-based normalization and structured extraction of concept entities and directed relations from instructional discourse streams.
- An incremental consolidation mechanism that merges semantically equivalent entities and filters low-information relations to control redundancy under repeated updates, while preserving evidence links for inspection.
- A reproducible replay-based evaluation and ablation suite (multi-seed aggregation) that characterizes online behavior in terms of final graph size, update latency, and LLM call counts.

II. Related Work

A. Classroom Discourse Analytics

Discourse analytics and discourse-centric learning analytics study how learning processes can be inferred from interaction

traces, including classroom discussion [1], [2]. A recurring requirement is that discourse-derived information should be summarized in forms that remain interpretable under classroom time constraints. This requirement is closely connected to the classroom orchestration and teacher-facing analytics literature, where compact representations support monitoring and reflection during activities [3]–[5]. Most discourse analytics pipelines, however, are developed and evaluated in offline settings and do not explicitly address online maintenance under repeated updates.

NLP-based approaches have been surveyed as practical tools for educational discourse modeling, while noting persistent challenges such as domain sensitivity, annotation cost, and evaluation reliability [7], [8]. More recently, LLMs have been evaluated for classroom dialogue analysis, indicating potential but also underscoring the need for careful constraints and validation [12]. In contrast to offline analysis and post-hoc summarization, our focus is the *systems problem* of sustaining a compact representation under continuous updates, where redundancy control and provenance become first-order design requirements.

B. Knowledge Graphs in Education

Knowledge graphs provide a general framework for representing entities and relations and have been widely used to integrate heterogeneous knowledge sources [13]. In educational applications, knowledge graphs have been constructed from learning resources to support downstream tasks such as recommendation and assessment; KnowEdu is a representative system [14]. Graph-based learner modeling has also leveraged concept graphs as structured priors, for example in graph-based knowledge tracing where concept dependencies are encoded as edges [15]. These lines of work typically assume batch construction or the availability of a predefined concept structure.

Graph representations have also been used to support qualitative inspection of knowledge structure derived from text. For example, [16] proposed a graphical interface for exploring knowledge structures extracted from text, and concept map generation pipelines have identified concepts and linking phrases to visualize semantic structure [17]. While these approaches demonstrate the utility of graph-based representations for inspection, they commonly operate on offline corpora and do not emphasize incremental maintenance under a discourse stream. Our setting differs in that the representation must be refreshed repeatedly under latency constraints, where uncontrolled redundancy directly degrades usability.

C. Extraction from Noisy Discourse

Spoken discourse contains disfluencies, fragments, and segmentation ambiguity; utterance-level processing is common in dialogue and discourse analysis [18], [19]. Information extraction from such data is challenged by noise and propagation of upstream errors. LLMs offer a flexible mechanism for normalization and structured extraction of entities and relations with limited task-specific training [9]–[11]. In educational contexts, LLM-based analysis has been studied, but reliability and evaluation remain central concerns [12].

In streaming settings, extraction quality alone is insufficient: repeated updates introduce redundancy, and representations can become rapidly cluttered without explicit consolidation and filtering. This motivates integrated designs that combine structured extraction with incremental entity consolidation, relation filtering, and evidence linking so that the resulting graph remains compact and inspectable over time.

III. System Overview

We model instructional discourse as a time-ordered stream of utterances. At update step t, the system processes a newly formed batch B_t and maintains an evolving knowledge graph $G_t = (V_t, E_t)$, where nodes represent consolidated concept entities and edges represent directed semantic relations extracted from discourse.

The prototype is implemented as an online loop with a fixed update cadence. Each update step consists of: (1) batch formation and storage of utterances with metadata (time, speaker/channel), (2) LLM-based normalization and schema-constrained extraction of concept entities and relations, (3) incremental consolidation (entity merging and relation filtering) to control redundancy, and (4) graph update and visualization with evidence links for inspection. The pipeline is topic-agnostic in the sense that it relies on strict output schemas and similarity-based consolidation; adaptation to specific subjects is performed through prompt templates and filter lists rather than retraining.

Figure 1 summarizes the end-to-end flow and highlights two design choices that are central in streaming settings: (i) the update loop enforces bounded work per step through batch formation, and (ii) extracted structures retain provenance links to supporting utterances for inspection. An optional retrieval channel can inject course-material context into the LLM stage, but the experiments in this paper focus on the core online loop.

IV. Online Knowledge Graph Construction

This section describes the online update pipeline that converts a discourse stream into an evolving knowledge graph. The system is designed to (i) bound work per update step, (ii) control redundancy under repeated updates, and (iii) preserve evidence links for inspection.

A. Batch Formation

Incoming utterances are grouped into update batches B_t using a fixed cadence (every Δ seconds) and an optional maximum count per batch. This design bounds computation and supports repeated refresh under online constraints. For multi-speaker or multi-channel streams, the system may use an alternating policy (a fixed number of utterances per channel per update) to prevent a single channel from dominating the incremental graph.

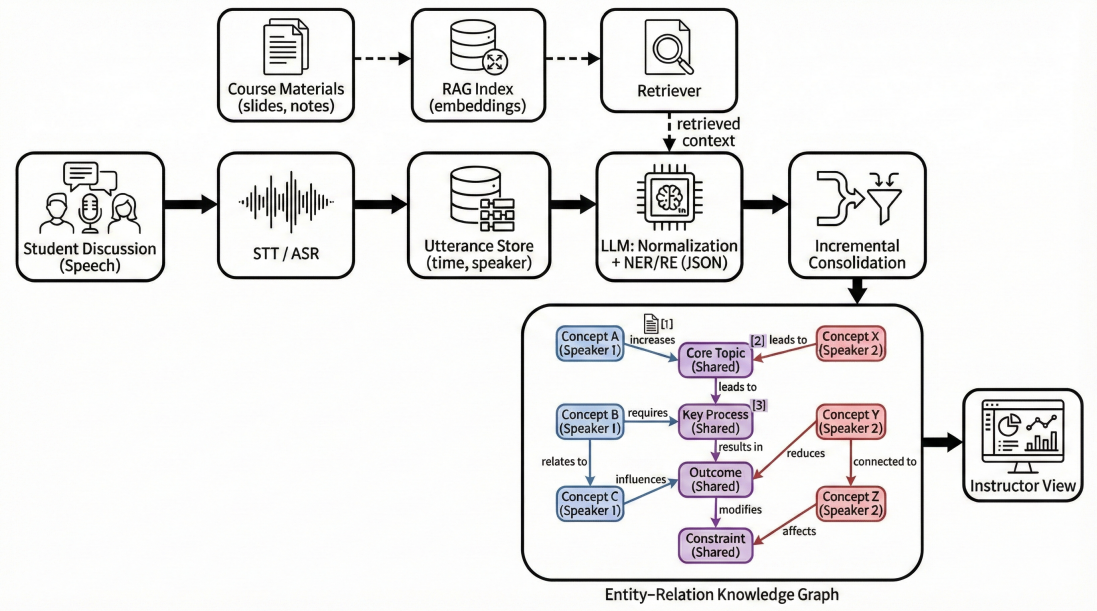


Fig. 1: End-to-end pipeline for streaming knowledge graph construction from classroom discourse. The optional course-material channel (top) provides retrieved context to the LLM stage, while the main online loop (bottom) constructs and updates an entity–relation knowledge graph for instructor-facing inspection.

B. Normalization, Entity Recognition, and Relation Extraction

At each update step, the system applies a two-stage LLM workflow: (i) normalization and (ii) structured extraction.

Normalization. Given B_t, the model rewrites ill-formed utterances into minimally well-formed sentences while preserving semantic intent. The normalization stage targets common artifacts of spontaneous discourse (fragments, repairs, informal phrasing) and produces normalized sentences $\hat{B}_t$ with provenance (channel and time). Normalized sentences are stored to support incremental processing and evidence linking.

Concept entities. From $\hat{B}_t$, the system identifies salient *concept entities*. Here, "entities" refer to domain-relevant concept phrases rather than conventional NER types (e.g., person/location). Entities are constrained to short noun phrases to support stable graph nodes under repeated updates.

Directed relations. The system extracts directed semantic relations among entities and represents each relation as a triple (h, r, t). Relation predicates are constrained to concise directional phrases (e.g., maximum three words) to reduce ambiguity and facilitate online consolidation. Generic predicates (e.g., *is*, *has*, *relates to*) are discouraged at extraction time and may be removed during filtering.

Output schema and validation. Extraction returns a machine-parseable structure:

$$\texttt{entities} = [e_1, e_2, \ldots], \quad \texttt{relations} = [(h, r, t), \ldots].$$

Outputs are schema-validated and failures are logged. In our evaluation suite, we did not observe JSON parse failures across runs.

C. Incremental Consolidation and Relation Filtering

Streaming extraction produces redundant entities due to lexical variation and repeated mentions. The system therefore consolidates entities and filters relations incrementally.

Candidate generation. For each newly extracted entity, merge candidates are generated using (i) lexical normalization (case folding, plural/singular normalization, and unambiguous abbreviations) and (ii) embedding-based similarity against existing canonical entities. Candidate generation limits verification cost under streaming constraints.

Optional verification and canonicalization. Ambiguous candidates can be verified using bounded local context (e.g., recent sentences) to avoid over-merging distinct concepts. When a merge is applied, the system assigns a canonical label and rewires incident relations to the canonical node. Canonical labels are selected using brevity and conventional usage.

Relation filtering. Relations are normalized to enforce consistent direction and reduce duplication. Low-information predicates are filtered to control clutter. When multiple relations occur between the same node pair within a short window, the system retains a reduced set of representative relations to preserve readability under repeated updates.

D. Graph Update and Rendering

After consolidation, new nodes and edges are appended to G_t. The visualization encodes provenance (e.g., channel-specific vs. shared concepts) and emphasizes salient nodes using a composite centrality score.

We compute a weighted sum of PageRank, degree centrality, and betweenness centrality on the current snapshot [20], [21]:

$$\text{score}(v) = w_1 \,\text{PR}(v) + w_2 \,\text{Deg}(v) + w_3 \,\text{Bet}(v), \quad (1)$$

Algorithm 1 Online update for streaming knowledge graph construction

Input: new batch B_t (utterances with timestamps, channel ids)
Output: updated graph $G_t = (V_t, E_t)$
$\hat{B}_t \leftarrow \text{NORMALIZE}(B_t)$
$(\mathcal{E}_t, \mathcal{R}_t) \leftarrow \text{EXTRACTENTITIESANDRELATIONS}(\hat{B}_t)$
Generate merge candidates via lexical rules and embedding similarity
Optionally verify and apply merges; canonicalize entity labels
Normalize/filter relations; rewire relations to canonical entities
Update V_t and E_t incrementally; compute salience scores
Render updated graph with provenance-aware styling and evidence links

where $w_1 + w_2 + w_3 = 1$ and each centrality is linearly rescaled to $[0, 1]$ per snapshot before aggregation. In our experiments, we use $(w_1, w_2, w_3) = (0.60, 0.25, 0.15)$ as a default. For efficiency, we compute betweenness centrality only when the number of nodes is at most 250; otherwise we set $\text{Bet}(v) = 0$. Node size and label emphasis follow $\text{score}(v)$, and nodes/edges retain links to supporting utterances for inspection.

V. Prototype Evaluation

This section evaluates the online behavior of the prototype under a controlled replay stream. We focus on feasibility-oriented metrics (final graph size, end-to-end update latency, and LLM call counts) and study component contributions via ablations.

A. Replay Protocol

We construct a replay stream from an IBM Project Debater corpus [22] using the topic *"We should ban cosmetic surgery"*. Utterances are processed using a fixed-size alternating batch policy (a small number of utterances per channel per update), resulting in 14 update steps under the default configuration. To account for stochastic variation, we repeat each variant with three random seeds.

In this corpus, labels indicate whether a sentence contains an argumentative statement for the topic rather than encoding pro/con stance. Accordingly, the channel identifiers in our replay are treated as *synthetic channels* for balanced batching and provenance-aware visualization, not as ground-truth debate stances.

B. Qualitative Evolution

Fig. 2 shows three snapshots of the evolving knowledge graph at early, mid, and late update steps (steps 3, 8, and 14). The snapshots illustrate how new concepts enter the graph as the stream progresses, how previously introduced concepts persist, and how incremental consolidation helps control redundancy under repeated updates.

C. Ablation Suite

We evaluate the contribution of major components through an ablation suite: (i) disabling normalization (`no_normalization`), (ii) disabling LLM merge verification while keeping embedding-based candidate generation (`no_llm_merge_verify`), (iii) disabling relation filtering (`no_relation_filter`), and (iv) disabling pruning in the visualization stage (`no_pruning`).

Fig. 3 summarizes final graph size, update latency, and total LLM call counts (mean±std over seeds). Fig. 4 provides a qualitative view of relation quality by comparing frequently observed predicates with and without relation filtering.

D. Measurements

We measure the following online metrics under replay:

- **Final graph size:** number of canonical nodes and edges after consolidation and filtering at the end of the replay stream.
- **Update latency:** end-to-end time per update step (normalization, extraction, consolidation, rendering).
- **LLM cost proxy:** total LLM call counts per run (normalization, extraction, and optional verification).
- **Compactness proxy:** compression ratio $C = M_{\text{raw}}/|V|$, where M_{raw} is the total number of extracted entity mentions before consolidation and $|V|$ is the number of canonical entities in the final graph.

The experimental logs additionally store auxiliary metrics (e.g., per-step growth traces, predicate statistics, and evidence support sizes), enabling deeper analysis beyond the scope of this compact report.

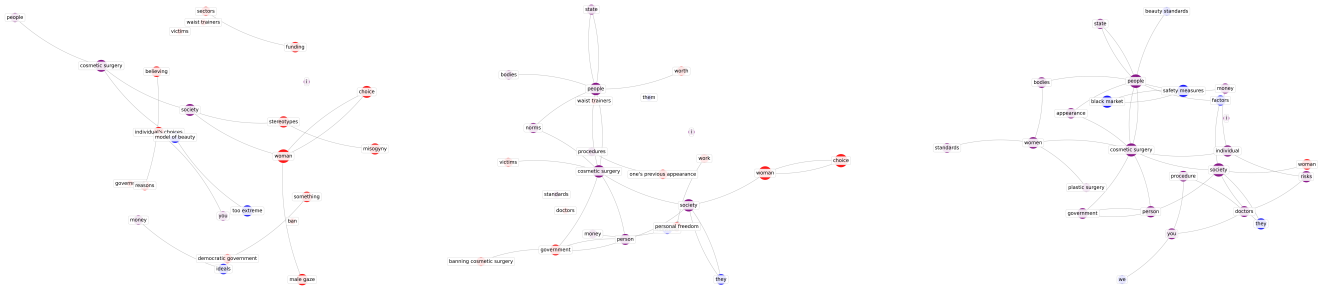
E. Key Observations

Table I summarizes the ablation results, and Fig. 3 visualizes the corresponding trends. First, normalization is the dominant cost lever. Disabling normalization reduces both end-to-end update latency and the number of LLM calls, and it yields a smaller final graph. We treat `no_normalization` as an efficiency-oriented ablation, since reduced normalization can also lower semantic coverage.

Second, disabling LLM-based merge verification (`no_llm_merge_verify`) reduces total LLM calls with little change in latency. This suggests that normalization and extraction dominate runtime in the current configuration. The smaller graph under embedding-only consolidation indicates more aggressive merging, which improves compactness but may increase over-merging risk.

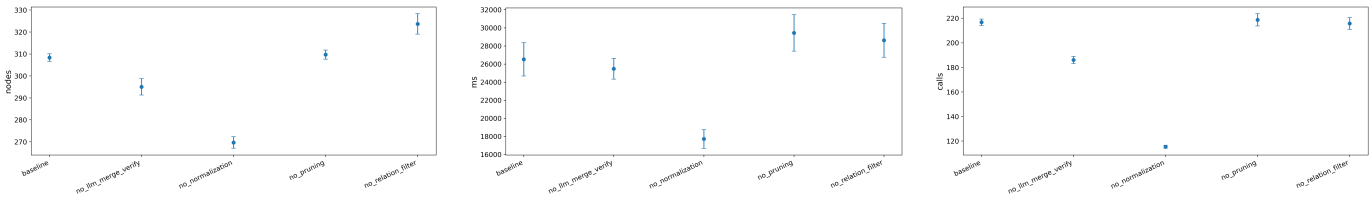
Third, relation filtering is important for readability under repeated updates. When filtering is disabled, the final graph becomes denser and latency increases. Fig. 4 shows that filtering suppresses generic predicates and shifts the distribution toward more informative relations, reducing clutter in the rendered graph.

Finally, disabling pruning (`no_pruning`) increases latency without materially changing the stored graph size, consistent with pruning primarily affecting visualization-stage cost rather than extraction and consolidation.



(a) Early stage. (b) Mid stage. (c) Late stage.

Fig. 2: Replay-based demonstration of time-evolving knowledge graph snapshots under the topic *"We should ban cosmetic surgery"* at early/mid/late update steps (step 3/8/14).

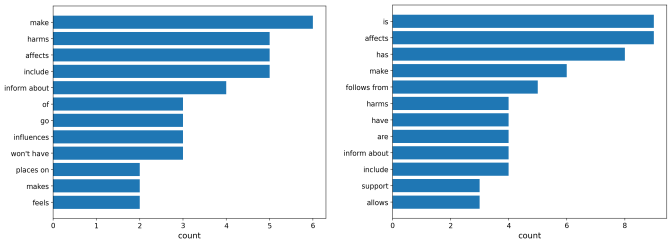


(a) Final nodes (mean±std). (b) Update latency (mean±std). (c) LLM calls (mean±std).

Fig. 3: Ablation suite summary (mean±std across seeds 41/42/43).

TABLE I: Ablation summary (mean±std across seeds). Latency: seconds per update. Calls: total LLM calls per run. Compr: raw entity mentions (pre-consolidation) divided by final canonical entities.

Variant	Final Nodes	Final Edges	Lat (s)	Calls (total)	Compr
`baseline`	308.3 ± 1.7	310.3 ± 3.4	26.5 ± 1.8	216.7 ± 2.6	1.45 ± 0.00
`no_llm_merge_verify`	295.0 ± 3.7	310.3 ± 4.0	25.5 ± 1.1	186.0 ± 2.9	1.53 ± 0.01
`no_normalization`	269.7 ± 2.6	260.7 ± 2.9	17.7 ± 1.0	115.3 ± 1.2	1.41 ± 0.02
`no_pruning`	309.7 ± 2.1	310.3 ± 3.1	29.4 ± 2.0	218.7 ± 5.0	1.45 ± 0.02
`no_relation_filter`	323.7 ± 4.6	334.7 ± 5.2	28.6 ± 1.9	215.7 ± 4.8	1.43 ± 0.01



(a) `baseline` (b) `no_relation_filter`

Fig. 4: Top predicates with and without relation filtering.

VI. Limitations and Future Work

This work reports a prototype pipeline and a controlled replay-based evaluation rather than a classroom-wide impact study. First, extraction quality depends on the stability of normalization and schema-constrained output; errors can propagate to consolidation. Second, consolidation introduces a trade-off between compactness and fidelity, and different thresholds and verification policies may change the resulting graph structure. Third, our replay uses a debate corpus and synthetic channels for controlled streaming; evaluating authentic classroom audio transcripts and validating whether channel-specific and shared concept patterns align with instructional interpretations remain important next steps. Finally, deployment in real instructional settings requires careful handling of consent, privacy, and data retention.

Beyond the debate replay setting, the pipeline is applicable to other instructional discourse streams where utterances arrive continuously and compactness is needed for inspection. Future work includes evaluation on classroom and tutoring data, teacher-in-the-loop studies, and systematic adaptation of prompts and filtering configurations across subject domains.

VII. Conclusion

We presented a prototype pipeline for streaming knowledge graph construction and visualization from classroom discourse using LLM-based normalization and structured extraction. The system maintains a compact evolving graph through incremental entity consolidation and relation filtering, while preserving evidence links for inspection. Using a reproducible replay-based setup, we characterized online behavior through final

graph size, update latency, LLM call counts, and an ablation suite aggregated across three seeds.

ACKNOWLEDGMENT

This work was partially supported by the Tech Incubator Program for Startup (TIPS) funded by the Korean government (MSS) (RS-2023-00303969) and by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2025S1A5C3A03022652).

REFERENCES

[1] S. Knight and K. Littleton, "Discourse-centric learning analytics: Mapping the terrain," *Journal of Learning Analytics*, vol. 2, no. 1, pp. 185–209, 2015.

[2] C. P. Rosé, "Discourse analytics," in *Handbook of Learning Analytics*. Society for Learning Analytics Research (SoLAR), 2017.

[3] P. Dillenbourg, "Design for classroom orchestration," *Computers & Education*, vol. 69, pp. 485–492, 2013.

[4] K. Verbert, S. Govaerts, E. Duval, J. L. Santos, F. Van Assche, G. Parra, and J. Klerkx, "Learning dashboards: An overview and future research opportunities," *Personal and Ubiquitous Computing*, vol. 18, no. 6, pp. 1499–1514, 2014.

[5] B. A. Schwendimann, M. J. Rodríguez-Triana, A. Vozniuk, L. P. Prieto, M. Shirvani Boroujeni, A. Holzer, D. Gillet, and P. Dillenbourg, "Perceiving learning at a glance: A systematic literature review of learning dashboard research," *IEEE Transactions on Learning Technologies*, vol. 10, no. 1, pp. 30–41, 2017.

[6] M. E. Webb, D. Prasse, M. Phillips, D. M. Kadijevich, C. Angeli, A. Strijker, A. A. Carvalho, B. B. Andresen *et al.*, "Challenges for it-enabled formative assessment of complex 21st century skills," *Technology, Knowledge and Learning*, vol. 23, pp. 441–456, 2018.

[7] N. Dowell and V. Kovanović, "Modeling educational discourse with natural language processing," in *Handbook of Learning Analytics*, 2nd ed. Society for Learning Analytics Research (SoLAR), 2022, pp. 105–119.

[8] R. Ferreira-Mello, M. A. D. Ferreira, A. Pinheiro, E. d. B. Costa, and C. Romero, "Text mining in education," *WIREs Data Mining and Knowledge Discovery*, vol. 9, no. 6, p. e1332, 2019.

[9] T. B. Brown, B. Mann, N. Ryder *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901.

[10] L. Ouyang, J. Wu, X. Jiang *et al.*, "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, 2022, arXiv:2203.02155.

[11] OpenAI, "GPT-4 technical report," 2023, technical report. [Online]. Available: https://cdn.openai.com/papers/gpt-4.pdf

[12] Y. Long, H. Luo, and Y. Zhang, "Evaluating large language models in analysing classroom dialogue," *npj Science of Learning*, vol. 9, no. 1, p. 60, 2024.

[13] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, S. Kirrane, J. E. Labra Gayo, R. Navigli, A. Polleres *et al.*, "Knowledge graphs," *ACM Computing Surveys*, vol. 54, no. 4, pp. 71:1–71:37, 2021.

[14] P. Chen, Y. Lu, V. W. Zheng, and X. Chen, "Knowedu: A system to construct knowledge graph for education," *IEEE Access*, vol. 6, pp. 31 553–31 563, 2018.

[15] H. Nakagawa, Y. Iwasawa, and Y. Matsuo, "Graph-based knowledge tracing: Modeling student proficiency using graph neural network," in *Proceedings of the 2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI '19)*, 2019, pp. 156–163.

[16] K. Kim, "Graphical interface of knowledge structure: A web-based research tool for representing knowledge structure in text," *Technology, Knowledge and Learning*, vol. 24, no. 1, pp. 89–95, 2019.

[17] C. Z. Aguiar, D. Cury, and A. Zouaq, "Automatic construction of concept maps from texts," in *Proceedings of the Seventh International Conference on Concept Mapping*, Tallinn, Estonia, 2016.

[18] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed., 2023, draft. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/

[19] H. H. Clark and E. F. Schaefer, "Contributing to discourse," *Cognitive Science*, vol. 13, no. 2, pp. 259–294, 1989.

[20] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Proceedings of the 7th International World Wide Web Conference (WWW7)*, 1998, pp. 107–117.

[21] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[22] S. Mirkin, M. Jacovi, T. Lavee, H.-K. Kuo, S. Thomas, L. Sager, L. Kotlerman, E. Venezian, and N. Slonim, "A recorded debating dataset," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), 2018.