

Resumo Técnico Comparativo MAIC & SINKT

Eduardo Prasniewski

18 de novembro de 2025

1 Anotações

1.1 MAIC

MAIC (Massive AI-empowered Course), é um novo paradigma para a educação online proposto para evoluir além do modelo estático "um vídeo para N alunos" dos MOOCs tradicionais. O objetivo principal do MAIC é equilibrar **escalabilidade** (servir muitos alunos) com **adaptabilidade** (fornecer uma experiência de aprendizagem personalizada). Ele atinge isso construindo uma "N agentes para 1 aluno" sala de aula dinâmica, onde um único aluno interage com uma equipe de agentes conduzidos por IA.

Arquitetura de Duas Fases

A estrutura MAIC é implementada combinando dois sistemas distintos: Slide2Lecture para preparação de curso offline e SimClass para entrega de curso em tempo real.

Fase 1 — Pipeline de Ensino Offline (Slide2Lecture): Este é o motor de criação do curso. Em vez de um humano gastar semanas gravando um vídeo, a estrutura Slide2Lecture atua como um pipeline de pré-processamento. Um professor fornece seus slides de aula brutos (por exemplo, um arquivo PowerPoint). O Slide2Lecture então usa LLMs multimodais para analisar este conteúdo (tanto texto quanto visual) e gera automaticamente um "roteiro de curso" completo e estruturado. Este roteiro é uma lista sequencial de ações de ensino, incluindo não apenas o conteúdo da aula (ReadScript), mas também quando mudar de slide (ShowFile) e quando questionar o aluno (AskQuestion).

Fase 2 — Ambiente de Aprendizagem Online (SimClass): Esta é a "sala de aula interativa" que o aluno experimenta. A estrutura SimClass toma o "roteiro de curso" estruturado gerado pelo Slide2Lecture como sua entrada. Em seguida, ela executa uma simulação multiagente

em tempo real. O núcleo do SimClass é o seu **Controlador de Sessão**, um meta-agente que dirige o fluxo da aula. Ele instrui o **Agente Professor** de IA a entregar o roteiro, enquanto também gerencia um conjunto de **Agentes Colegas de Classe** (como o Pensador Profundo ou Mente Inquisitiva) para simular uma discussão colaborativa e ao vivo. Quando o aluno interage, o Controlador de Sessão gerencia dinamicamente a conversação, misturando perfeitamente o roteiro pré-planejado com o diálogo adaptativo em tempo real.

1.1.1 Fluxo de Trabalho de Ensino

Fase de Leitura (Read Stage) O processo começa com os *slides* do curso, \mathcal{P} , definidos como um conjunto de páginas individuais:

$$\mathcal{P} = \{P_i\}_{1 \leq i \leq |\mathcal{P}|} \quad (1)$$

Estes são processados em recursos de aprendizagem inteligentes estruturados, $\hat{\mathcal{P}}$:

$$\hat{\mathcal{P}} = \{\langle P_i, D_i, K_j \rangle\}_{1 \leq i \leq |\mathcal{P}|, 1 \leq j \leq |\hat{\mathcal{P}}|} \quad (2)$$

1.1 Extração de Conteúdo do Slide Uma função f_T^1 extrai o conteúdo textual (P_i^t) e visual (P_i^v) de cada página P_i .

$$f_T^1 : P_i \rightarrow \langle P_i^t, P_i^v \rangle \quad (3)$$

1.2 Extração de Estrutura Uma segunda função f_T^2 gera uma descrição abrangente D_i a partir do conteúdo extraído.

$$f_T^2 : \langle P_i^t, P_i^v \rangle \rightarrow D_i \quad (4)$$

Fase de Planejamento (Plan Stage) Esta fase define as ações de ensino, \mathcal{T} , como uma tupla de tipo e valor.

$$\mathcal{T} = (\text{tipo}, \text{valor}) \quad (5)$$

O tipo de ação é selecionado a partir de um conjunto pre-definido:

$$\text{tipo} \in \{\text{ShowFile}, \text{ReadScript}, \text{AskQuestion}\} \quad (6)$$

Cada ação \mathcal{T}_n está associada a um conteúdo específico $\hat{P}_{\mathcal{T}}$.

$$\langle \mathcal{T}_n, \hat{P}_{\mathcal{T}} \rangle \quad (7)$$

1.1.2 Fluxo de Trabalho de Aprendizagem

Papéis da Turma e Conteúdo O ambiente de aprendizagem é definido por um conjunto de papéis da turma $\hat{\mathcal{R}}$ (por exemplo, TI, EC, CM) e o roteiro instrucional sequenciado C .

$$\hat{\mathcal{R}} = \{r_i\}^{|\hat{\mathcal{R}}|} \quad (8)$$

$$C = \{c_1, \dots, c_t\} \quad (9)$$

Controlador de Sessão O Controlador de Sessão gerencia o ambiente de aprendizagem em tempo real.

Recetor de Estado da Turma O histórico de diálogo H_t é a união de todas as falas u_i de todos os agentes a_j até o tempo t .

$$H_t = \bigcup (u_i^{a_j})^t \quad (10)$$

O estado geral da turma S_t é uma função dos materiais ensinados P_t , do histórico H_t e dos papéis ativos $\hat{\mathcal{R}}$.

$$S_t = \{P_t, H_t \mid \hat{\mathcal{R}}\}, \text{ onde } P_t \subseteq \mathcal{P} \quad (11)$$

Função do Agente Gerenciador A lógica central do controlador é uma função $f_{\mathcal{L}}$ que pega o estado atual S_t e seleciona o próximo agente a_t e a ação de ensino \mathcal{T} a ser executada.

$$f_{\mathcal{L}} : S_t \rightarrow (a_t, \mathcal{T}) \mid a_t \in \mathcal{A}, \mathcal{T}_n \leftarrow \mathcal{T} \quad (12)$$

1.2 SINKT

O artigo apresenta o SINKT, um modelo inovador para Rastreamento de Conhecimento (Knowledge Tracing - KT),

que é a tarefa de prever se um aluno responderá corretamente à próxima pergunta com base em seu desempenho anterior. O SINKT busca resolver os problemas de *cold-start* (partida a frio), escassez de dados (data sparsity) e falta de estrutura dos modelos tradicionais de KT.

1.2.1 Definição do problema

Consideramos um conjunto de alunos S , um conjunto de perguntas Q e um conjunto de conceitos C . O histórico de aprendizagem de um aluno $s \in S$ é registrado como uma sequência

$$R_s = \{(q_1, r_1), (q_2, r_2), \dots, (q_T, r_T)\}$$

, onde $q_t \in Q$ é a pergunta respondida na etapa de tempo t , e $r_t \in \{0, 1\}$ é a correção da resposta (1 para correto, 0 para incorreto).

Dado o histórico de aprendizagem R_s de um aluno e uma nova pergunta q_{T+1} , o objetivo da tarefa de Rastreamento de Conhecimento (KT) é prever a probabilidade de o aluno responder corretamente à nova pergunta. Isso é denotado como:

$$p(r_{T+1} = 1 \mid R_s, q_{T+1}) \quad (13)$$

Na tarefa de KT tradicional (transdutiva), o conjunto de perguntas nos dados de treinamento e nos dados de teste são os mesmos ($Q_{\text{trein}} = Q_{\text{teste}}$). Este trabalho também aborda a **tarefa de KT indutiva**, onde o modelo deve fazer previsões para novas perguntas não vistas durante o treinamento ($Q_{\text{trein}} \neq Q_{\text{teste}}$).

Esta seção detalha a estrutura SINKT, que é composta por quatro componentes principais: um Codificador de Informação Textual (TIEnc), um Codificador de Informação Estrutural (SIEnc), um Codificador de Estado do Aluno e um Preditor de Resposta.

1.2.2 Codificador de Informação Textual (TIEnc)

Para capturar a informação semântica de conceitos e perguntas, um Modelo de Linguagem Pré-treinado (PLM) é usado como codificador semântico. O texto puro de cada conceito c_i e pergunta q_i é inserido no PLM para adquirir seus respectivos vetores de representação:

$$x_i^c = \text{PLM}_c(\text{TEXT}(c_i)) \in \mathbb{R}^{d_t} \quad (14)$$

$$x_i^q = \text{PLM}_q(\text{TEXT}(q_i)) \in \mathbb{R}^{d_t} \quad (15)$$

Aqui, d_t representa a dimensão de codificação do PLM. Esses vetores, x_i^c e x_i^q , servem como as entradas semânticas iniciais para os módulos subsequentes.

1.2.3 Codificador de Informação Estrutural (SIEnc)

Um codificador de grafo heterogêneo multi-camadas é projetado para capturar a informação estrutural do grafo conceito-pergunta. Este codificador processa três tipos de relacionamentos usando três Redes de Atenção Gráfica (GATs) distintas:

1. **GAT Conceito-Pergunta (C-Q):** Funde representações de conceitos vizinhos para uma pergunta alvo.
2. **GAT Conceito-Conceito (C-C):** Integra informações de conceitos vizinhos para um conceito alvo.
3. **GAT Pergunta-Conceito (Q-C):** Integra informações de perguntas vizinhas para um conceito alvo.

Seja $\mathcal{N}_{c_i}^q$ o conjunto de perguntas vizinhas para o conceito c_i , $\mathcal{N}_{c_i}^c$ os conceitos vizinhos para o conceito c_i , e $\mathcal{N}_{q_i}^c$ os conceitos vizinhos para a pergunta q_i .

O mecanismo de atenção para o GAT Conceito-Pergunta (C-Q), que atualiza as representações das perguntas, é definido como:

$$\alpha_{i,j}^{cq} = \frac{\exp(\text{LeakyReLU}(a_{cq}^T(x_i^q \oplus x_j^c)))}{\sum_{c_k \in \mathcal{N}_{q_i}^c} \exp(\text{LeakyReLU}(a_{cq}^T(x_i^q \oplus x_k^c)))} \quad (16)$$

$$e_i^{cq} = \sum_{c_j \in \mathcal{N}_{q_i}^c} \alpha_{i,j}^{cq} \cdot (W_{cq} x_j^c) \quad (17)$$

Onde W_{cq} é a matriz de peso de agregação, a_{cq} é o vetor de peso de atenção, e \oplus denota concatenação.

De forma similar, os GATs Conceito-Conceito (C-C) e Pergunta-Conceito (Q-C) são definidos como:

$$\alpha_{i,j}^{cc} = \frac{\exp(\text{LeakyReLU}(a_{cc}^T(x_i^c \oplus x_j^c)))}{\sum_{c_k \in \mathcal{N}_{c_i}^c} \exp(\text{LeakyReLU}(a_{cc}^T(x_i^c \oplus x_k^c)))} \quad (18)$$

$$e_i^{cc} = \sum_{c_j \in \mathcal{N}_{c_i}^c} \alpha_{i,j}^{cc} \cdot (W_{cc} x_j^c) \quad (19)$$

$$\alpha_{i,j}^{qc} = \frac{\exp(\text{LeakyReLU}(a_{qc}^T(x_i^c \oplus x_j^q)))}{\sum_{q_k \in \mathcal{N}_{c_i}^q} \exp(\text{LeakyReLU}(a_{qc}^T(x_i^c \oplus x_k^q)))} \quad (20)$$

$$e_i^{qc} = \sum_{q_j \in \mathcal{N}_{c_i}^q} \alpha_{i,j}^{qc} \cdot (W_{qc} x_j^q) \quad (21)$$

Para enfatizar a informação semântica original, o SINKT introduz o conhecimento saltitante (*jumping knowledge*). A representação do nó na l -ésima camada do codificador é:

$$x_i^{c(l)} = \text{ReLU}(W_c x_i^{c(l-1)} + e_i^{cc} + e_i^{qc}) \quad (22)$$

$$x_i^{q(l)} = \text{ReLU}(W_q x_i^{q(l-1)} + e_i^{cq}) \quad (23)$$

Onde W_c e W_q são matrizes de peso treináveis. A entrada na camada 0 é inicializada pela saída do TIEnc:

$$x_i^{c(0)} = x_i^c, \quad x_i^{q(0)} = x_i^q \quad (24)$$

Após k camadas, obtemos os vetores de representação finais $\tilde{c}_i \in \mathbb{R}^d$ e $\tilde{q}_i \in \mathbb{R}^d$.

1.2.4 Codificador de Estado do Aluno

Para modelar o histórico de aprendizagem do aluno, as interações são transformadas para o nível de conceito. Para uma pergunta q_t com um conjunto de conceitos \mathcal{C}_{q_t} , a representação no nível de conceito u_t é a média dos vetores de conceito:

$$u_t = \frac{1}{|\mathcal{C}_{q_t}|} \sum_{c_i \in \mathcal{C}_{q_t}} \tilde{c}_i \quad (25)$$

Para representar conjuntamente o item e a correção da resposta r_t , um vetor de interação $v_t \in \mathbb{R}^{2d}$ é introduzido:

$$v_t = \begin{cases} u_t \oplus \mathbf{0} & \text{se } r_t = 1 \\ \mathbf{0} \oplus u_t & \text{se } r_t = 0 \end{cases} \quad (26)$$

Onde $\mathbf{0} \in \mathbb{R}^d$ é um vetor nulo. Esta sequência é modelada usando uma Unidade Recorrente com Portões (GRU):

$$u_r = \sigma(W_r(v_t \oplus h_{t-1}) + b_r) \quad (27)$$

$$u_z = \sigma(W_z(v_t \oplus h_{t-1}) + b_z) \quad (28)$$

$$u_h = \tanh(W_h(v_t \oplus (u_r * h_{t-1})) + b_h) \quad (29)$$

$$h_t = (1 - u_z) * u_h + u_z * h_{t-1} \quad (30)$$

O estado oculto $h_t \in \mathbb{R}^d$ representa o estado de conhecimento do aluno na etapa de tempo t .

1.2.5 Predição de Resposta

A probabilidade de uma resposta correta y_{t+1} para a próxima pergunta q_{t+1} é prevista combinando o estado de conhecimento do aluno h_t , a representação da pergunta \tilde{q}_{t+1} e a representação no nível de conceito da pergunta u_{t+1} :

$$y_{t+1} = \sigma(W_p(h_t \oplus \tilde{q}_{t+1} \oplus u_{t+1}) + b_p) \quad (31)$$

Onde $W_p \in \mathbb{R}^{3d}$ e $b_p \in \mathbb{R}$ são parâmetros treináveis. O modelo é treinado minimizando a perda de entropia cruzada entre a previsão y_t e a resposta de verdade fundamental r_t :

$$\mathcal{L} = - \sum_{t=1}^T (r_t \log y_t + (1 - r_t) \log(1 - y_t)) \quad (32)$$

1.2.6 Discussões

A natureza estrutural e indutiva do SINKT possibilita diversas aplicações únicas em EdTech:

- **Design Curricular:** Gera automaticamente mapas de habilidades de pré-requisito (Árvores de Habilidades) usando LLMs para inferir o fluxo lógico entre novos conceitos.
- **Diagnóstico da Causa Raiz:** Rastreia erros de volta através do grafo para identificar lacunas de conhecimento fundamental (por exemplo, conceitos de pré-requisito não dominados) em vez de apenas o erro imediato.
- **Calibração Cold Start:** Estima instantaneamente a dificuldade e o escopo de conteúdo novo e não visto, analisando o texto da pergunta.

- **Tutoria Adaptativa:** Facilita a criação de problemas de prática diversos e únicos, fornecendo avaliação instantânea de questões novas, geradas por IA.

Comparação com BKT e Integração

- **Diferença:** O BKT é um modelo probabilístico e transdutivo que depende de dados de ID históricos, enquanto o SINKT é um modelo de *deep learning* indutivo que depende de vetores semânticos ricos e grafos de conceito explícitos.
- **Integração:** Eles podem ser usados em conjunto (conjunto). O método ideal é executar o BKT em paralelo e alimentar sua probabilidade de domínio interpretável como um **recurso de entrada adicional** na camada de previsão do SINKT, aproveitando a simplicidade do BKT sem estrangular o complexo estado vetorial do SINKT.

Treinamento do Modelo e Resolução de Cold Start

- **Componentes do Modelo:** O “Modelo SINKT” consiste em codificadores PLM reutilizáveis (por exemplo, BERT) e camadas centrais treinadas sob medida (GRU/GAT). Você deve obter a implementação e treinar as camadas centrais.
- **Estratégia de Treinamento:** O treinamento deve seguir um processo de duas etapas 1) **Pré treinamento Universal** em dados públicos agregados para ensinar a mecânica da aprendizagem, e 2) **Ajuste Fino Específico do Cliente** (opcional) usando os dados isolados do cliente (*Transfer Learning*) para especialização.
- **Cold Start do Cliente:** Para um novo curso de cliente sem histórico, o SINKT usa o TIEnc pré-treinado universalmente para generalizar a semântica da pergunta e depende do LLM para gerar automaticamente o grafo de pré-requisitos, permitindo que o modelo faça previsões informadas imediatamente.

2 Simulações

2.1 MAIC

Esta seção detalha a aplicação do MAIC, onde o aluno repete o mesmo erro mais de uma vez.

- **Cenário:** Erro Repetido no Conceito K01 (IndexError em Python).
- **Detalhe:** Erro na tentativa 2, associado à falha em acessar corretamente um índice dentro dos limites de uma lista (ex: tentar acessar o índice 10 em uma lista de 5 itens).
- **Gatilho do MAIC:** O sistema de agentes é acionado porque a falha é considerada de alta prioridade (erro conceitual básico) e a proficiência do aluno ainda não cruzou o limiar de domínio completo.

2.1.1 Resposta dos Agentes

Abaixo, a simulação da intervenção simultânea dos três agentes do MAIC, cada um com um papel pedagógico específico (Instrutivo, Estratégico e Emocional), conforme o modelo de sala de aula aumentada proposto pelo framework.

Tabela 1: Diálogo simulado

Agente	Mensagem e Papel (Resposta)
Tutor	“Olá! Notei que você teve dificuldade na questão de indexação. Lembre-se que em Python, a contagem começa em 0 e o último item é sempre <code>tamanho - 1</code> . Tentar acessar um índice maior causa o erro que você viu. Vamos tentar corrigir o código juntos?”
Conselho	“Percebi que você assistiu ao vídeo, mas errou o quiz logo em seguida. Talvez a teoria ainda esteja abstrata. Sugiro que você olhe este diagrama visual sobre como os índices funcionam na memória antes de prosseguir para o próximo exercício prático. Isso vai consolidar sua base.”
Amigo	“Nossa, eu sempre confundo esses índices também! É super normal errar isso no começo. O importante é que no exercício final você mandou super bem. Bora pro próximo que a gente consegue!”

2.1.2 Objetivo pedagógico e resultado esperado

- **Objetivo Pedagógico Central (Scaffolding):** Garantir a correção da falha conceitual de indexação, utilizando a intervenção coordenada dos três agentes.
- **Três Focos de Intervenção:**
 1. **Cognitivo/Conceitual:** O **Tutor** fornece a correção técnica imediata.
 2. **Metacognitivo/Estratégico:** O **Conselho** ajusta a estratégia de estudo, recomendando uma revisão de recurso antes da prática.
 3. **Afetivo/Emocional:** O **Amigo** mantém a motivação, normalizando o erro e incentivando a continuidade.
- **Resultado Esperado:** Após esta intervenção direcionada, espera-se que o aluno realize uma nova tentativa com sucesso e permitindo o desbloqueio do próximo módulo (K02) pelo sistema.

2.2 Bayesian Knowledge Tracing (BKT)

Esta seção apresenta a simulação do rastreamento de conhecimento para o conceito K01 (Listas em Python), utilizando uma fórmula de atualização de domínio inspirada nos princípios do Bayesian Knowledge Tracing (BKT). A planilha visualiza a progressão do domínio do aluno ao longo de cinco tentativas, com base em diferentes tipos de atividades e seus respectivos pesos.

A fórmula de atualização aplicada é:

$$p_t = p_{t-1} + \alpha \cdot \beta \cdot (r_t - p_{t-1})$$

Onde α (taxa de aprendizado) é fixado em 0.30.

Tabela 2: Rastreamento de Domínio do Conceito K01: Simulação BKT Simplificada

Tentativa (t)	Tipo de Atividade	Resposta (r_t)	Peso (β)	Domínio Anterior (p_{t-1})	Domínio Atual (p_t)
0	Conhecimento prévio	-	-	-	0.50
1	Vídeo	1	0.3	0.50	0.55
2	Quiz (Fácil)	0	0.6	0.55	0.45
3	Leitura	1	0.2	0.45	0.48
4	Quiz (Médio)	1	0.8	0.48	0.60
5	Exercício de Código	1	1.0	0.60	0.72

Análise dos Resultados da Planilha:

A tabela, derivada dos cálculos apresentado detalha a evolução do domínio do aluno:

- Conhecimento Prévio (p_0):** O aluno inicia com um domínio predefinido de 0.50 (50%).
- Tentativa 1 (Vídeo):** Após assistir a um vídeo (resposta 1, peso 0.3), o domínio aumenta marginalmente para 0.55. O baixo peso reflete a natureza passiva da atividade.
- Tentativa 2 (Quiz Fácil):** Um erro em um quiz fácil (resposta 0, peso 0.6) causa uma queda significativa no domínio para 0.45, indicando uma lacuna fundamental.
- Tentativa 3 (Leitura):** A leitura de material (resposta 1, peso 0.2) eleva o domínio apenas ligeiramente para 0.48, pois a evidência de aprendizado é menos forte.
- Tentativa 4 (Quiz Médio):** Um acerto em um quiz de dificuldade média (resposta 1, peso 0.8) impulsiona o domínio para 0.60, mostrando que o aluno está começando a consolidar o conhecimento.
- Tentativa 5 (Exercício de Código):** O sucesso em um exercício prático de código (resposta 1, peso 1.0) resulta em um aumento substancial, levando o domínio para 0.72. Esta atividade, com o maior peso, é a evidência mais forte de proficiência.

Interpretação:

O resultado final de 0.72 (72%) para o domínio do conceito K01 indica que o aluno está progredindo e se aproxima de um nível de proficiência. No entanto, a trajetória mostra que atividades passivas e erros em questões básicas têm um

impacto notável, enquanto atividades práticas e de maior peso são cruciais para a consolidação do conhecimento. Este ponto de domínio pode ser um gatilho para o MAIC intervir, seja para reforço ou para avançar para o próximo conceito.

3 Conceitos-chaves (gerado por IA)

3.1 MAIC

1. **Mudança de Paradigma:** O MAIC (Curso Massivo Capacitado por IA) propõe uma transição do modelo tradicional "Um Vídeo para N Alunos" dos MOOCs para um modelo personalizado "N Agentes para 1 Aluno", visando equilibrar escalabilidade com adaptabilidade.
2. **Arquitetura de Sistema Duplo:** A estrutura é construída sobre dois sistemas distintos, mas integrados: Slide2Lecture para preparação de curso offline e SimClass para o ambiente de aprendizagem online e interativo.
3. **Geração Automatizada de Conteúdo:** O *pipeline* Slide2Lecture utiliza Grandes Modelos de Linguagem (LLMs) multimodais para extrair automaticamente conteúdo visual e textual de *slides* de aula brutos, convertendo-os em roteiros de ensino estruturados e agendas sem a necessidade de ajuste fino humano.
4. **Eficiência de Custo e Tempo:** O MAIC reduz significativamente os recursos necessários para a criação de cursos, cortando o tempo e o custo de semanas e milhares de dólares (para produção de vídeo) para meros minutos e custos negligenciáveis por curso.
5. **Simulação de Sala de Aula Multiagente:** O ambiente *SimClass* simula uma sala de aula realista implantando múltiplos agentes de IA, incluindo um Professor, um Assistente de Ensino e diversos Colegas de Classe (por exemplo, Pensador Profundo, Piadista da Turma), para fomentar a aprendizagem social.
6. **Controle Dinâmico de Sessão:** Um novo meta-agente "Controlador de Sessão" gerencia o fluxo da aula em tempo real. Ele decide dinamicamente qual agente deve falar em seguida com base no histórico de diálogo e no estado de ensino atual, em vez de seguir um roteiro rígido e linear.
7. **Ações de Ensino Heterogêneas:** O sistema formaliza o ensino em ações discretas (por exemplo, *ReadScript*, *ShowFile*, *AskQuestion*), permitindo uma

entrega de conteúdo estruturada, mas flexível, que imita padrões pedagógicos humanos.

8. **Experiência de Aprendizagem Interativa:** Ao contrário da observação passiva de vídeos, o MAIC permite que os alunos interrompam, façam perguntas e se envolvam em discussões profundas com agentes a qualquer momento, com o sistema fornecendo *feedback* imediato e sensível ao contexto.
9. **Engajamento Comportamental:** Estudos empíricos mostram que os alunos no ambiente MAIC exibem altos níveis de engajamento proativo, com porções significativas da interação envolvendo questionamento e discussão ativos, em vez de consumo passivo.
10. **Validação no Mundo Real:** A estrutura foi implementada e avaliada em cursos universitários reais (por exemplo, "Rumo à Inteligência Artificial Geral"), demonstrando que agentes conduzidos por LLMs podem entregar conhecimento complexo de forma eficaz e manter alta qualidade de ensino.

3.2 SINKT

1. **Capacidade Indutiva:** O SINKT é o primeiro modelo de Rastreamento de Conhecimento capaz da *tarefa indutiva*, o que significa que ele pode prever o desempenho do aluno em novas perguntas e conceitos que não estavam presentes durante o treinamento, sem a necessidade de retreinamento.
2. **Integração de LLMs:** Ele utiliza Grandes Modelos de Linguagem (LLMs) de duas maneiras distintas: gerando relações estruturais de mundo aberto entre conceitos (usando GPT-4) e codificando o significado semântico de texto puro (usando PLMs como BERT ou Vicuna).
3. **Codificador de Informação Textual (TIEnc):** Ao contrário dos modelos tradicionais que dependem de *embeddings* de ID aleatórios, o SINKT usa um Modelo de Linguagem Pré-treinado (PLM) para criar representações vetoriais semânticas a partir do texto original das perguntas e conceitos.

4. **Grafo com Consciência Estrutural:** O modelo constrói um grafo heterogêneo contendo três tipos de arestas: Conceito-Pergunta, Pergunta-Conceito e as novas arestas Conceito-Conceito geradas pelo GPT-4 para capturar relações de pré-requisito.
5. **Codificador de Informação Estrutural (SIEnc):** Ele emprega uma Rede de Atenção Gráfica (GAT) multi-camadas para agregar informações da vizinhança, permitindo que o modelo aprenda com a topologia complexa do grafo conceito-pergunta.
6. **Conhecimento Saltitante (Jumping Knowledge):** Para evitar a perda de riqueza semântica durante a propagação do grafo, o SINKT usa uma técnica de "conhecimento saltitante" que preserva as representações originais baseadas em texto junto com as aprimoradas pelo grafo.
7. **Solução para Cold-Start:** Ao depender de dados textuais e estruturais em vez de apenas do histórico de interações, o SINKT aborda de forma eficaz o problema de "cold start" (partida a frio), onde novos conteúdos educacionais carecem de dados de interação suficientes do aluno.
8. **Robustez à Escassez de Dados:** Experimentos mostram que o SINKT mantém alto desempenho mesmo quando os dados de treinamento são limitados (por exemplo, com menos alunos), superando significativamente as linhas de base em cenários de dados esparsos.
9. **Modelagem do Estado do Aluno:** Ele utiliza uma Unidade Recorrente com Portões (GRU) para rastrear a evolução temporal do estado de conhecimento de um aluno com base na sequência de suas interações passadas.
10. **Desempenho de Ponta (State-of-the-Art):** O SINKT supera consistentemente 12 modelos de KT transdutivos existentes (incluindo DKT, AKT e GKT) em quatro conjuntos de dados do mundo real (ASSIST09, ASSIST12, Junyi e Programming).