From Lexical to Entity-Level: Predicting Individual Knowledge Boundaries via Knowledge Graph Label Propagation

Shuyi Yu **Yusuke Sakai** **Hidetaka Kamigaito** **Taro Watanabe**
Nara Institute of Science and Technology (NAIST), Japan
yu.shuyi.yx3@naist.ac.jp
{sakai.yusuke.sr9, kamigaito.h, taro}@is.naist.jp

Abstract

Personalized reading assistance requires identifying specific concepts a reader finds unfamiliar. While prior work on personalized difficulty prediction has focused on the word or jargon level, a reader's knowledge boundary is more naturally defined over entities grounded in knowledge bases. In this paper, we introduce a novel task of predicting entity-level familiarity for individual readers and propose a graph-based framework for prediction. The system extracts Wikipedia entities from a document, constructs a document-specific graph in the entity embedding space, selects maximally informative seed entities for reader annotation through active learning, and propagates these sparse labels to the remaining entities. We evaluate the framework on 90 Wikipedia articles across five knowledge domains, and find that our method outperforms GPT-4.1-mini and performs comparably to Qwen2.5-72B-Instruct-AWQ without requiring LLM inference. Ablation experiments confirm the contribution of each component. We further analyze the impact of incorporating structured knowledge and popularity-guided directed propagation, finding that both yield domain-dependent effects rather than universal gains. These results lay a practical foundation for constructing reading assistance systems that adapt to individual knowledge backgrounds. Our code and data are available at https://github.com/yyushuyui/entity-familiarity-prediction.

1 Introduction

Personalized reading assistance requires identifying the specific concepts a reader finds unfamiliar. When a reader encounters an unknown entity, such as a technical term like MOSFET or a cultural concept like Peking Opera, it immediately distracts their attention and forces them to look up definitions elsewhere, thus interrupting the reading process. This mismatch between what the content assumes a reader knows and what they actually know varies across individuals (Guo et al., 2024). As illustrated in Figure 1, two readers with different cultural backgrounds can exhibit markedly different knowledge boundaries over the same article: entities that are obvious to one reader may be entirely unknown to another. Existing methods for identifying difficult terms mainly rely on corpus-level familiarity indicators rather than the needs of individual readers (Guo et al., 2024), so the same explanations are offered to all readers, overwhelming experts with redundant information while leaving novices under-served. The fundamen-

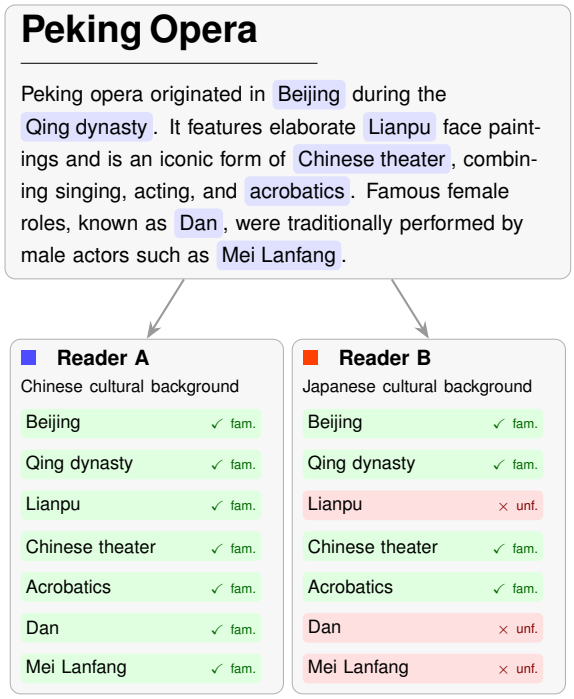


Figure 1: Two readers encounter the same article but exhibit different knowledge boundaries. Reader A (Chinese cultural background) is familiar with culture-specific entities such as Lianpu, Dan and Mei Lanfang, while Reader B (Japanese cultural background) finds them unfamiliar. Our task is to predict this individual-level knowledge boundary over entities.

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 1763–1778
November 6-10, 2026 ©2026 Association for Computational Linguistics

tal bottleneck is not the generation of explanations, but the upstream problem of predicting what an individual reader already knows.

Prior work on personalized difficulty prediction has focused almost exclusively on the word or jargon level (Gooding and Tragut, 2022; Guo et al., 2024), where predicting difficulty for individual readers remains challenging (Nohejl et al., 2024). The closest methodological precedent is Ehara et al. (2014), who formalize vocabulary prediction as graph-based active learning over word graphs built from corpus frequency and domain information. However, a reader's knowledge boundary is more naturally defined over entities, which are structured concepts grounded in knowledge bases, rather than surface-level lexical items; knowledge graphs provide rich structured facts about entities that text-only representations rarely incorporate (Zhang et al., 2019). Adaptive learning research likewise models learners' knowledge together with the structure of knowledge items (Liu et al., 2019). To our knowledge, no prior work predicts entity-level familiarity for individual readers across domains by leveraging knowledge graph structure.

In this paper, we introduce a novel task of predicting entity-level familiarity for individual readers and propose a graph-based framework for prediction built on a four-stage pipeline. First, the system extracts Wikipedia entities from a document via entity linking using the ReFinED model (Ayoola et al., 2022). Second, it constructs a document-specific subgraph by establishing a fully connected structure among these extracted entities (referred to as core entities) with Gaussian-kernel edge weights derived from Wikipedia2Vec embeddings (Yamada et al., 2020) and further enriches the subgraph through k-nearest-neighbor expansion in the same embedding space. Third, we select a small set of informative entities using V-optimality-based active learning on the graph Laplacian (Ji and Han, 2012), and use them for reader annotation. Fourth, we spread these sparse labels to all remaining entities via label propagation.

We evaluate the framework on 90 Wikipedia articles spanning five knowledge domains with annotators from diverse academic and cultural backgrounds. Each reader provides complete familiarity annotations for all entities in the corpus; the system uses only 20 seed labels per reader to predict the rest, and its predictions are compared with the reader's own annotations to assess how well it recovers each reader's knowledge boundary. Our method significantly outperforms GPT-4.1-mini and performs comparably to Qwen2.5-72B-Instruct-AWQ, a 72B-parameter LLM, without requiring LLM inference. Ablation experiments confirm the contribution of edge weighting, the fully connected core graph, and active seed selection, with Gaussian kernel weighting proving most critical. We further analyze the impact of incorporating structured relations from YAGO 4.5 (Suchanek et al., 2024) and popularity-guided directed propagation, finding that both yield domain-dependent effects rather than universal gains. These results demonstrate that graph-based label propagation from minimal reader input is an effective approach to personalized entity familiarity prediction, laying a practical foundation for next-generation reading assistance systems that adapt to individual knowledge backgrounds.

2 Related Work

Personalized Lexical Complexity. Complex Word Identification and Lexical Complexity Prediction (LCP) traditionally treat difficulty as a universal text property (Shardlow et al., 2022; Nohejl et al., 2024). Recent work challenges this: Gooding and Tragut (2022) show word complexity is reader-dependent, Ide et al. (2023) construct a Japanese LCP dataset for non-native readers with first-language-specific annotations, Nohejl et al. (2024) report that individual-level prediction remains difficult, and Guo et al. (2024) introduce personalized jargon identification with 11 computer science researchers. We extend this line from lexical items to knowledge-graph-grounded entities, where relational structure provides signals unavailable to word-level models.

Graph-Based Active Learning and Knowledge Tracing. Ehara et al. (2014) formalize vocabulary prediction as graph-based active learning over word graphs built from corpus frequency and domain information. We build on this paradigm but operate over entities and adopt V-optimality (Ji and Han, 2012) for query selection. Our task also parallels knowledge tracing (Corbett and Anderson, 1994; Piech et al., 2015), which models learner mastery from interaction sequences. However, unlike educational settings where a learner answers questions sequentially and the system updates its estimate after each response, reading comprehension provides no such interaction history. We therefore frame the problem as a single-round prediction task: given

a small set of seed annotations, the system infers familiarity for all remaining entities over a static entity subgraph.

Knowledge Graph-Based User Modeling. Knowledge graphs have been widely used to enhance user and entity representations. KGAT (Wang et al., 2019) captures high-order relational signals through attentive propagation, ERNIE (Zhang et al., 2019) integrates entity knowledge into pre-trained language models, and label propagation (Zhou et al., 2003) provides a principled framework for spreading sparse signals through graph structure. Our pipeline combines Wikipedia2Vec kNN subgraphs (Yamada et al., 2020) with Gaussian-kernel label propagation in a lightweight architecture. Unlike prior KG-based methods that model item preferences for recommendation, our framework models a reader's knowledge state, that is, whether they are familiar with a given concept.

Preference Learning. Jiang et al. (2024) survey human preference learning for LLMs, and Xie et al. (2025) identify user-modeling-based methods as one of three categories of personalized alignment techniques, alongside training-time and inference-time methods. A reader's familiarity with an entity is itself a preference signal, as it determines whether an explanation is needed. Our work provides the modeling infrastructure for this signal at the entity level, bridging the gap between word-level complexity prediction and system-level personalized assistance.

3 Methodology

When a reader opens a document, they may be familiar with some of the concepts mentioned but unfamiliar with others. Our goal is to automatically identify which entities in a document a specific reader already knows, without having to ask about every single one. We use Wikipedia entities as our unit of concepts, as they provide unique, structured representations of real-world knowledge that go beyond surface-level words.

Formally, given a document $\mathcal{D}$ and a reader r, let $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ be the set of Wikipedia entities identified in $\mathcal{D}$ through entity linking. For each entity e_i, we predict a binary label $y_i \in \{0, 1\}$, where $y_i = 1$ indicates familiarity and $y_i = 0$ indicates unfamiliarity. Following prior work on personalized word, vocabulary, and jargon difficulty (Gooding and Tragut, 2022; Ehara et al., 2014; Guo et al., 2024), we treat familiarity as binary.

Querying the reader about every entity is impractical when n is large. Instead, we select a small subset $\mathcal{Q} \subset \mathcal{E}$ of $q \ll n$ entities for the reader to annotate and propagate these labels to the remaining entities. Our approach exploits the relational structure of entity familiarity: for a given reader, semantically similar entities tend to share familiarity status. This is the smoothness assumption behind label propagation (Zhu and Ghahramani, 2002; Zhou et al., 2003), also adopted by Ehara et al. (2014) for vocabulary prediction. We formalize this intuition by constructing a weighted entity graph and performing label propagation over it.

As illustrated in Figure 2, our pipeline consists of four stages: (1) entity extraction (§3.1), (2) document-specific subgraph construction (§3.2), (3) representative entity sampling (§3.3), and (4) label propagation and prediction (§3.4). The concrete hyperparameter settings for each stage are provided in the Implementation Details of §4.1.

3.1 Entity Extraction

Entity extraction identifies the key concepts in $\mathcal{D}$ that a reader may or may not be familiar with, forming the basis for familiarity prediction. To extract these concepts, we apply the entity linking model ReFinED (Ayoola et al., 2022) to identify all entity mentions in $\mathcal{D}$, yielding a set of **core entities** $\mathcal{E} = \{e_1, \ldots, e_n\}$. Each core entity e_i is mapped to its unique Wikipedia identifier, providing access to a pre-trained entity embedding $\boldsymbol{v}_i \in \mathbb{R}^d$ from Wikipedia2Vec (Yamada et al., 2020). Entities that are recognized by the linker but absent from the embedding index are excluded from graph construction and directly classified as unfamiliar.

3.2 Document-Specific Subgraph Construction

To predict familiarity, we need a structure that captures how the concepts in a document relate to one another for a given reader. Rather than operating over the full Wikipedia2Vec entity space, which contains millions of entities and would be computationally prohibitive, we build a graph that is specific to the document at hand. This design choice offers two additional advantages: it keeps the graph small and tractable, and it avoids introducing a large number of irrelevant entities that would add noise to label propagation.

Concretely, we construct a document-specific

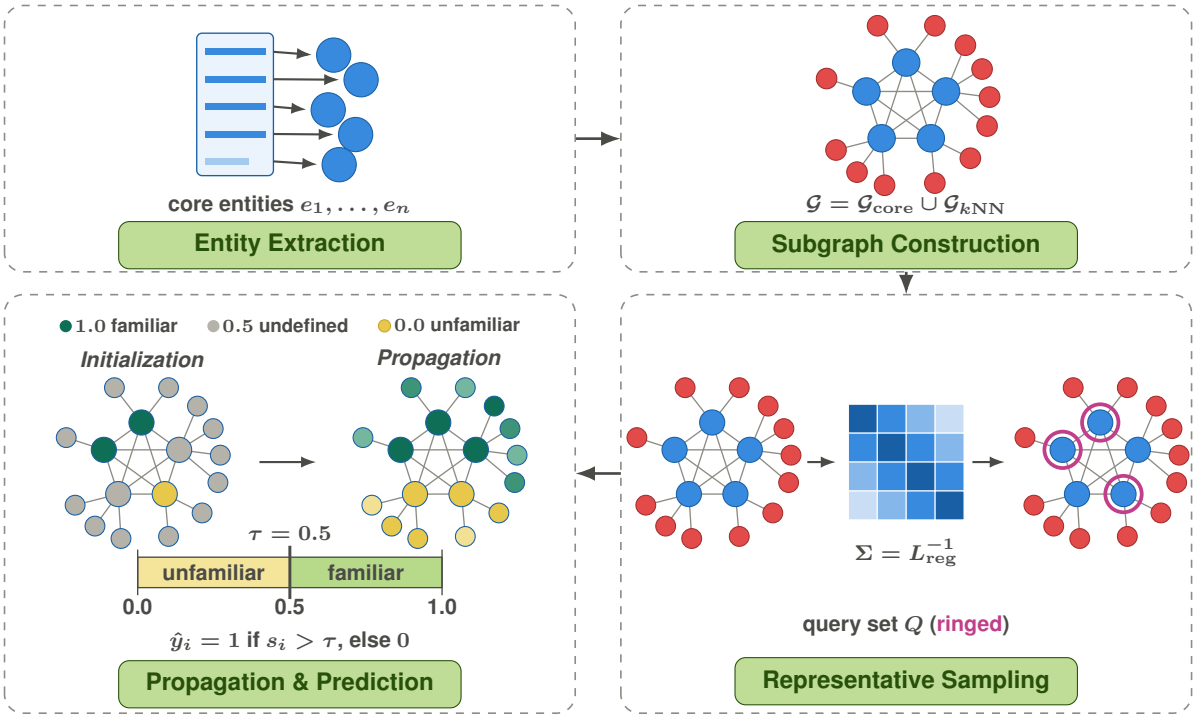


Figure 2: **Overview of the entity familiarity prediction framework. (1) Entity Extraction** (§3.1): Wikipedia entities in the document are identified and designated as **core entities**. **(2) Subgraph Construction** (§3.2): **core entities (blue)** form a fully connected clique, augmented by ***k*NN expansion entities (red)**. **(3) Representative Sampling** (§3.3): the V-optimality criterion greedily selects q representative core entities as the query set $\mathcal{Q}$ (ringed; $q = 3$ in this example) for reader annotation. **(4) Propagation & Prediction** (§3.4): the three annotated seeds are initialized to 1.0 (familiar) or 0.0 (unfamiliar), and all other nodes to 0.5; iterative weighted averaging then propagates their labels to the whole graph. Nodes scoring above $\tau = 0.5$ are predicted as familiar (**green**) and the rest as unfamiliar (**yellow**).

entity subgraph $\mathcal{G}$ with vertex set $\mathcal{V}$ in two steps. We first build a complete subgraph $\mathcal{G}_{\text{core}}$ over the n core entities, with vertex set $\mathcal{V}_{\text{core}} = \mathcal{E}$. Every pair of distinct core entities is connected by an undirected edge, ensuring that familiarity signals can propagate directly between any two entities mentioned in the document.

To enrich the graph with broader semantic context, we then expand the vertex set using k-nearest-neighbor retrieval in the Wikipedia2Vec embedding space, which contains vectors for over 4 million entities. For each core entity $e_i \in \mathcal{V}_{\text{core}}$, we retrieve the k closest entities by Euclidean distance. If a retrieved neighbor e' is not already in $\mathcal{V}_{\text{core}}$, it is added to an auxiliary set $\mathcal{V}_{k\text{NN}}$ and connected to every core entity that retrieved it; we denote this expansion subgraph by $\mathcal{G}_{k\text{NN}}$, so that $\mathcal{G} = \mathcal{G}_{\text{core}} \cup \mathcal{G}_{k\text{NN}}$ (Figure 2). The final vertex set is $\mathcal{V} = \mathcal{V}_{\text{core}} \cup \mathcal{V}_{k\text{NN}}$. This topology preserves semantic context while keeping the graph tractable, as a fully connected graph over all vertices would incur $\mathcal{O}(|\mathcal{V}|^2)$ edges.

All edges in $\mathcal{G}$ are weighted by a Gaussian kernel over Wikipedia2Vec embeddings, a common choice in label propagation (Zhu and Ghahramani, 2002):

$$w_{ij} = \exp\left(-\frac{\|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2}{\sigma^2}\right), \qquad (1)$$

where $\|\boldsymbol{v}_i - \boldsymbol{v}_j\|$ is the Euclidean distance between entity embeddings and σ is the bandwidth parameter. This formulation assigns high weights to semantically similar entity pairs, reflecting the intuition that closely related concepts tend to share familiarity status for a given reader.

3.3 Representative Entity Sampling

Given the graph $\mathcal{G}$ and an annotation budget of q queries, we select a subset $\mathcal{Q}$ of core entities that are most informative for mapping the reader's knowledge boundary, using the V-optimality criterion (Ji and Han, 2012), which selects nodes that maximally reduce the total predictive variance over the graph. In the following, we use u and v to index

entities in $\mathcal{V}$, with the understanding that candidate queries are restricted to core entities in $\mathcal{E}$ defined in §3.1. We model entity familiarity as a Gaussian random field on $\mathcal{G}$, i.e., a joint Gaussian distribution whose precision matrix is the regularized graph Laplacian defined below, so that strongly connected entities are expected to have similar familiarity. Let $\mathbf{W}$ be the weighted adjacency matrix and $\mathbf{D}$ the diagonal degree matrix with $D_{uu} = \sum_v W_{uv}$. The normalized graph Laplacian is:

$$\mathbf{L} = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-1/2}. \tag{2}$$

Adding a regularization parameter $\gamma > 0$ yields:

$$\mathbf{L}_{\text{reg}} = \mathbf{L} + \gamma\mathbf{I}, \tag{3}$$

and the covariance matrix is $\mathbf{\Sigma} = \mathbf{L}_{\text{reg}}^{-1}$, where Σ_{uv} captures the predictive coupling between entities u and v: larger values indicate that observing one entity's label substantially reduces uncertainty about the other.

We initialize $\mathbf{\Sigma}^{(0)} = \mathbf{\Sigma}_C$ (the submatrix restricted to core entities) and greedily select q entities to minimize total remaining variance. At each step t, we choose the entity v^* whose observation yields the largest reduction in total variance over all core entities:

$$v^* = \arg\max_{v \in \mathcal{E} \setminus \mathcal{Q}^{(t-1)}} \frac{\sum_u \left(\Sigma_{uv}^{(t-1)}\right)^2}{\Sigma_{vv}^{(t-1)}}, \tag{4}$$

where the sum runs over all core entities u and $\mathcal{Q}^{(t-1)}$ is the set of entities selected in the previous $t-1$ steps. The covariance matrix is then updated via the Schur complement to condition on the selected entity:

$$\mathbf{\Sigma}^{(t)} = \mathbf{\Sigma}^{(t-1)} - \frac{\boldsymbol{c}_{v^*}\boldsymbol{c}_{v^*}^\top}{\Sigma_{v^*v^*}^{(t-1)}}, \tag{5}$$

where $\boldsymbol{c}_{v^*}$ is the column of $\mathbf{\Sigma}^{(t-1)}$ corresponding to v^*. After q rounds, the selected entities $\mathcal{Q} = \{v_1^*, \ldots, v_q^*\}$ are presented to the reader for binary annotation (familiar/unfamiliar).

3.4 Label Propagation and Prediction

Given the reader's annotations on $\mathcal{Q}$, we propagate labels to all remaining entities via iterative weighted averaging, following the label propagation framework of Zhu and Ghahramani (2002). Each node v receives an initial familiarity score:

$$s_v^{(0)} = \begin{cases} 1.0 & \text{if } v \text{ is annotated as familiar,} \\ 0.0 & \text{if } v \text{ is annotated as unfamiliar,} \\ 0.5 & \text{otherwise (unlabeled).} \end{cases} \tag{6}$$

At each iteration t, every unlabeled node v updates its score as the weighted average of its neighbors:

$$s_v^{(t+1)} = \frac{\sum_{u \in N(v)} w_{vu} \cdot s_u^{(t)}}{\sum_{u \in N(v)} w_{vu}}, \tag{7}$$

where $N(v)$ is the set of neighbors of v and w_{vu} is the Gaussian edge weight from Equation (1). Annotated nodes in $\mathcal{Q}$ retain their initial scores throughout all iterations, acting as boundary conditions that anchor the propagation.

After T iterations, each core entity e_i has a continuous score $s_i \in [0, 1]$. We produce binary predictions using a threshold τ:

$$\hat{y}_i = \begin{cases} 1 \text{ (familiar)} & \text{if } s_i > \tau, \\ 0 \text{ (unfamiliar)} & \text{if } s_i \leq \tau. \end{cases} \tag{8}$$

We use strict inequality so that entities remaining at the default score of 0.5, those unreachable from any annotated node, are classified as unfamiliar. This reflects a conservative design principle: in a reading assistance setting, providing an unnecessary explanation is preferable to omitting one for an unknown entity.

4 Experiments

4.1 Experimental Setup

Human Evaluation. We curate 90 English Wikipedia pages spanning three knowledge dimensions: (i) Common Knowledge: 30 pages of everyday topics; (ii) Specialized Knowledge: 15 Computer Science and 15 Semiconductor Technology pages; and (iii) Cultural Knowledge: 15 Chinese Culture and 15 Japanese Culture pages. The specialized and cultural subcategories are designed to probe whether the system can distinguish readers with different domain expertise and cultural backgrounds. The complete page list is provided in Appendix C.

We recruit eight annotators: six with a computer science background and two with a semiconductor engineering background. Among them, six have a Chinese cultural background and two have a Japanese background. We deliberately control this

composition so that domain expertise and cultural background, the two factors our evaluation targets, are clearly identifiable. We extract entities from the entire corpus using ReFinED (Ayoola et al., 2022) and aggregate them into a single deduplicated annotation list. To disambiguate entities that may share surface forms (e.g., "Java" referring to the programming language, the island, or the coffee), we retrieve a short description for each entity via the Wikipedia REST API.[1] This description, typically the first sentence of the entity's Wikipedia abstract, is presented alongside the entity name during annotation, ensuring annotators judge the correct referent. Each participant independently labels every entity in the annotation list as "Familiar" or "Unfamiliar," yielding a complete ground-truth knowledge boundary per reader. The annotation list contains 10,130 entities, so each reader provides 10,130 labels. Because our goal is to predict each individual reader's knowledge boundary, every reader must label all entities, unlike typical crowdsourcing, where each annotator labels only a subset and the labels are aggregated into a single ground truth. Such complete annotation is costly, which keeps the number of participants small, as in prior studies on personalized vocabulary and jargon prediction that collected complete labels from 15 learners (Ehara et al., 2014) and 11 researchers (Guo et al., 2024). During evaluation, the system sees only 20 seed labels per reader per article and predicts the labels of the remaining entities, which we then compare with the reader's own annotations to measure how accurately the system recovers that reader's knowledge boundary. Details of the annotation protocol are in Appendix C.

Baselines. We compare our framework against three baselines. The *All-Familiar* baseline labels every entity as familiar, the more frequent label overall. It uses no information about the reader and thus shows what can be achieved from the label distribution alone. The two LLM baselines, *GPT-4.1-mini* (OpenAI, 2025) and *Qwen2.5-72B-Instruct-AWQ* (Qwen et al., 2025), receive the same 20 seed annotations as our method and are prompted to predict the familiarity of the remaining entities via in-context learning. Each unlabeled entity is presented to the model together with its Wikipedia description to provide the same disambiguation context given to human annotators. The full prompts are provided in Appendix F. To ensure a fair comparison, all methods operate under the same per-reader seed budget and are evaluated on the same entities, namely the core entities not used as seeds.

[1] https://en.wikipedia.org/api/rest_v1/

Stage	Parameter	Value
Entity Linking	Model	ReFinED
kNN Expansion	k (neighbors)	10
Edge Weights	σ (bandwidth)	0.75
Sampling	γ (regularization)	0.01
	q (annotation budget)	20
Propagation	T (iterations)	20
Prediction	τ (threshold)	0.5

Table 1: Hyperparameter settings.

Implementation Details. We report the hyperparameter settings for our graph-based framework described in §3; a full summary is given in Table 1. In the graph construction stage, the kNN expansion retrieves k=10 neighbors per core entity from the Wikipedia2Vec embedding space using Euclidean distance, and edge weights are computed with a Gaussian kernel of bandwidth σ=0.75. In the active seed selection stage, the V-optimality criterion uses a Laplacian regularization of γ=0.01 to select an annotation budget of q=20 seed entities per reader. In the label propagation stage, propagation runs for T=20 iterations and a decision threshold of τ=0.5 is applied to produce binary predictions.

We conduct sensitivity analyses over all five hyperparameters (k, σ, γ, q, T) in Appendix D. The framework is stable across all tested values: k, γ, and T each vary by fewer than 0.3 F1 points, while q and σ show slightly wider but still modest variation, about 3 points and under 5 points, respectively, confirming that performance is robust to the specific hyperparameter settings.

Model sizes and computational costs are reported in Appendix E.

4.2 Results and Analysis

Table 2 reports accuracy, precision, recall, and F1 with familiar as the positive class. Because labeling every entity as familiar already yields a high F1, we also report macro-F1: we compute F1 once with familiar and once with unfamiliar as the positive class and average the two, so a method must also recognize unfamiliar entities to score well. Our method significantly outperforms *GPT-4.1-mini* in both F1 and macro-F1 and performs on par with *Qwen2.5-72B-Instruct-AWQ*. It obtains the highest accuracy and macro-F1, although the macro-F1 gap to the 72B model is not significant. The 72B model

Method	Overall					Per-Domain F1				
	Acc	Prec	Rec	F1	Macro-F1	Common	Chinese	Japanese	CS	Semi.
All-Familiar	0.553	0.553	1.000	0.694	0.347	0.622	0.748	0.717	0.697	0.688
GPT-4.1-mini	0.661	**0.778**	0.459	0.568	0.615	0.458	0.610	0.687	0.574	0.513
Qwen2.5-72B-Instruct-AWQ	0.695	0.669	**0.807**	**0.725**	0.649	**0.692**	0.752	**0.749**	**0.699**	**0.731**
Ours	**0.698**	0.690	0.720	0.702	**0.653**	0.665	**0.780**	0.746	0.691	0.626

Table 2: Main results across five knowledge domains. Overall is the mean of the five domain scores. Bold marks the best result among methods other than *All-Familiar*. Per-annotator breakdowns are in Appendix A.

	Cultural		Specialized	
	Chinese	Japanese	CS	Semi.
In-domain	0.800	0.865	0.737	0.557
Out-of-domain	0.718	0.706	0.554	0.650
ΔF1	+0.082	+0.159	+0.183	−0.093

Table 3: Alignment effect: mean F1 of our method when the reader's background matches the article domain (In-domain) vs. otherwise (Out-of-domain).

Variant	F1	ΔF1 (pp)
Full system	0.702	—
Random sampling	$0.687_{\pm 0.004}$	−1.5
Uniform weights	0.614	−8.7
Sparse core ($d_{\max}$=7)	0.671	−3.1
Sparse core ($d_{\max}$=8)	0.694	−0.8
Sparse core ($d_{\max}$=9)	0.699	−0.3

Table 4: Ablation study results. ΔF1 reports the difference from the full system in percentage points. Random sampling is averaged over 5 seeds.

reaches a significantly higher F1 (0.725 vs. 0.702) by labeling more entities as familiar: it correctly identifies more of the entities the reader knows (recall 0.807 vs. 0.720), but more of its familiar predictions are wrong (precision 0.669 vs. 0.690). *GPT-4.1-mini* errs in the opposite direction: it labels too few entities as familiar and misses more than half of the entities the reader knows (recall 0.459). Our method lies between the two LLMs, with precision and recall close to each other, and requires no LLM inference. The *All-Familiar* baseline obtains a deceptively high F1 only because it never misses a familiar entity, but it never recognizes an unfamiliar one, so its macro-F1 is only 0.347.

Our method performs best on the two cultural domains and worst on Semiconductors. Compared with *Qwen2.5-72B-Instruct-AWQ*, it is better on Chinese Culture, close on Japanese Culture and Computer Science, and behind on Common Knowledge and especially Semiconductors (0.626 vs. 0.731). Full per-domain results are in Appendix B.

Table 3 shows a clear alignment effect: our method achieves higher F1 when the reader's background matches the article domain, with the largest gain on Computer Science (+0.183). Semiconductors is the only exception (−0.093): its in-domain readers label relatively few entities as familiar, which leaves few familiar seeds for propagation.

Overall, these results show that propagating a few labels from each reader over the entity graph is an effective and efficient way to predict individual knowledge boundaries: with only 20 seed labels per reader and no LLM inference, our method significantly outperforms *GPT-4.1-mini* and performs comparably to a 72B-parameter LLM.

4.3 Ablation Study

Our framework relies on three design choices: V-optimality for seed selection (§3.3), Gaussian kernel edge weights, and a fully connected graph over the core entities (§3.2). Each could be replaced by a simpler alternative, namely random sampling, uniform weights, or a sparser graph, so we conduct ablation experiments to verify that each choice actually contributes to performance. Starting from the full system, we replace one component at a time with its alternative while keeping all other settings fixed (k=10, σ=0.75, γ=0.01, q=20, T=20). Table 4 summarizes the results.

V-optimality vs. Random Sampling. Our framework relies on graph-informed seed selection to choose the most representative entities for annotation. A natural question is whether this strategic selection genuinely outperforms simply querying random entities. We replace V-optimality with uniform random sampling from the core entities and report the mean over 5 seeds. Random sampling degrades F1 by 1.5 points (0.687 vs. 0.702), with low variance across seeds (±0.4 points). The improvement is driven by both higher precision (+0.6

Variant	Common	Chinese	Japanese	CS	Semi.	Overall
Full system	0.665	0.780	0.746	0.691	0.626	0.702
+ YAGO (α=0.5)	0.662	0.785	0.754	0.686	0.621	0.701
ΔF1	−0.3	+0.5	+0.8	−0.5	−0.6	0.0
+ Directed	0.594	0.726	0.660	0.610	0.597	0.637
ΔF1	−7.1	−5.4	−8.6	−8.2	−3.0	−6.5

Table 5: Effect of incorporating YAGO 4.5 edges and directed propagation. Values are per-domain F1.

points) and recall (+2.5 points), indicating that V-optimality selects seeds that better cover the knowledge boundary. The gain is consistent across all five domains, with the largest improvements in Common Knowledge (+1.9 points) and Chinese Culture (+2.1 points), where the entity space is larger and strategic selection matters more.

Gaussian Kernel vs. Uniform Weights. The Gaussian kernel gives larger weights to edges between entities that are closer in the embedding space, reflecting the assumption that closely related concepts tend to share familiarity status. To test whether this weighting is necessary, we set all edge weights to a constant $w_{ij} = 1$. This causes the largest drop among all variants (−8.7 F1 points), with precision falling from 0.690 to 0.590. With equal weights, an entity is influenced as much by unrelated entities as by closely related ones, so familiar labels spread to unrelated entities and more of the familiar predictions are wrong. The drop is largest in Common Knowledge (−17.4 F1 points), whose articles mention entities from many different topics, so equal weights let many unrelated entities affect each prediction.

Fully Connected Core vs. Sparse Core Graph. Our method connects every pair of core entities, so that labels can pass directly between any two entities in the document. An alternative is to connect only entity pairs whose embedding distance is below a threshold $d_{\max}$, which would reduce computational cost for documents with many entities. We test $d_{\max} \in \{7, 8, 9\}$, where a smaller threshold removes more edges. Performance degrades gradually as the graph becomes sparser: at $d_{\max}$=9 the drop is minimal (−0.3 points), but at $d_{\max}$=7 it reaches −3.1 points, mainly because fewer familiar entities are found (recall falls from 0.720 to 0.647). Thus, the direct edges between less similar core entities still carry useful label information, which indirect paths through other entities cannot fully replace. Since a single document contains a limited number of core entities, connecting all of them adds little computational cost, so we keep the fully connected graph.

5 Discussion

Our framework constructs entity graphs solely from Wikipedia2Vec embeddings and uses undirected weighted averaging for label propagation (§3). Two natural questions arise: (1) whether incorporating structured knowledge from an external knowledge base can improve the entity graph, and (2) whether restricting propagation to one direction can reduce noise from distant entities. Table 5 compares both extensions with the full system.

Incorporating Structured Knowledge. External knowledge bases such as YAGO 4.5 (Suchanek et al., 2024) encode taxonomic and relational structure (e.g., `subClassOf`, `isPartOf`) that Wikipedia2Vec embeddings may not fully capture. We augment edge weights with a binary YAGO adjacency signal $a_{ij} \in \{0, 1\}$, indicating whether entities e_i and e_j share at least one relation from a manually selected set of relation types (Appendix G):

$$w_{ij}^{\text{hybrid}} = \alpha \cdot w_{ij} + (1 - \alpha) \cdot a_{ij}, \qquad (9)$$

where $\alpha = 1.0$ recovers the embedding-only default. Setting α=0.5 yields effectively unchanged overall F1 (0.701 vs. 0.702). YAGO marginally helps cultural domains (Chinese +0.5, Japanese +0.8) but slightly hurts specialized ones (CS −0.5, Semiconductors −0.6), likely due to limited coverage introducing noisy connections. One possible explanation is that Wikipedia2Vec embeddings already encode much of this relational structure. Domain-adaptive graph construction that selectively incorporates YAGO for domains with strong coverage remains a promising direction.

Directed Propagation. Since every pair of core entities is connected, each unlabeled node also receives signals from distant core entities, which may

dilute propagation. We therefore make propagation directional, using Wikipedia pageview counts[2] to estimate how general an entity is, based on the intuition that knowing a niche concept implies familiarity with broader ones. Let $\text{pop}(e)$ denote the pageview count of entity e in January 2026. In the directed variant, node v aggregates only from less popular neighbors:

$$s_v^{(t+1)} = \frac{\sum_{u \in N^-(v)} w_{vu} \cdot s_u^{(t)}}{\sum_{u \in N^-(v)} w_{vu}}, \qquad (10)$$

where $N^-(v) = \{u \in N(v) \mid \text{pop}(u) < \text{pop}(v)\}$. This causes a substantial overall drop (-6.5 F1 points), consistent across all domains, with the largest degradation on Japanese Culture (-8.6) and Computer Science (-8.2). Semiconductors shows the smallest decline (-3.0), possibly because pageviews better reflect how general entities are in this domain. We attribute this failure to two factors: pageviews reflect public attention rather than how general an entity is, and one-way propagation removes many useful connections. Exploring better measures of generality, such as an entity's depth in a concept hierarchy, remains a promising direction.

6 Conclusion

We proposed a graph-based framework for predicting personalized entity familiarity. The pipeline extracts Wikipedia entities, constructs a document-specific subgraph weighted by embedding similarity, selects informative seeds through V-optimality, and propagates sparse annotations to all remaining entities. Evaluations across five knowledge domains show that the framework significantly outperforms GPT-4.1-mini and performs comparably to Qwen2.5-72B-Instruct-AWQ without requiring LLM inference. Ablation experiments support our current design, with Gaussian kernel weighting being the most critical component. Incorporating structured YAGO relations and directed propagation (§5) yields no universal gains. This work establishes a practical foundation for adaptive reading assistance that maps individual knowledge boundaries from minimal input. Integrating these predictions with personalized explanation generation is a natural next step; limitations and future directions are detailed in §7.

7 Limitations

This work predicts which entities are unfamiliar but does not address how to explain them. A complete reading assistance system requires generating personalized explanations tailored to each reader's knowledge profile. Integrating familiarity prediction with efficient, on-the-fly explanation generation is a natural next step.

Prediction quality is lowest in the Semiconductors domain (F1: 0.626), where *Qwen2.5-72B-Instruct-AWQ* performs clearly better (0.731). More generally, this LLM attains a higher positive-class F1 than our method in four of the five domains, mainly through higher recall, which suggests combining graph propagation with the knowledge of LLMs as a future direction. Neither extension in §5 closes this gap.

The system uses a uniform annotation budget (q=20) across all documents, with no mechanism to adapt based on document complexity or reader background. Although our sensitivity analysis (Appendix D) shows stable performance across $q \in [10, 30]$, the best budget may differ across documents and domains. Developing an adaptive budget selection strategy that accounts for document difficulty and domain specificity is a promising direction.

Our evaluation covers only English Wikipedia articles with East Asian participants. The Wikipedia2Vec embeddings we use are trained on English Wikipedia and may not reflect how readers in other linguistic or cultural communities relate concepts. Multilingual extension and more diverse participant recruitment are needed for broader applicability.

Finally, the framework depends on pre-trained Wikipedia2Vec embeddings. Incorporating dynamic signals or periodic resource updates could improve robustness to emerging entities and shifting reader interests.

8 Ethical Considerations

This study involved human participants who annotated their own entity familiarity. All participants provided explicit informed consent after being briefed on the study's objectives and their right to withdraw at any time. Since the task required participants to identify personal knowledge gaps,

[2]Wikipedia pageview data were aggregated in January 2026 via the official Wikimedia dumps: https://dumps.wikimedia.org/other/pageview_complete/2026/2026-01/. To maintain the integrity of personalized modeling, we employ user-based splits and ignore standard automated splits.

we took care to minimize potential performance anxiety by emphasizing that unfamiliarity with specific entities is a natural and expected outcome of the research, rather than a reflection of academic or cognitive ability. All collected data were strictly anonymized to ensure that no individual knowledge profiles could be traced back to personal identities. Participants were compensated fairly for their time at a rate consistent with institutional standards, regardless of task outcomes.

Beyond this study, predicted familiarity reveals what a person does not know and should be treated as personal information: systems built on our method should obtain user consent and should not use such predictions to evaluate or rank people. Incorrect predictions may also lead a system to skip explanations that a reader needs, so readers should still be able to request explanations for any entity.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP26K00002.

References

Tom Ayoola, Shubhi Tyagi, Joseph Fisher, Christos Christodoulopoulos, and Andrea Pierleoni. 2022. ReFinED: An efficient zero-shot-capable approach to end-to-end entity linking. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 209–220, Hybrid: Seattle, Washington + Online. Association for Computational Linguistics.

Albert T. Corbett and John R. Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278.

Yo Ehara, Yusuke Miyao, Hidekazu Oiwa, Issei Sato, and Hiroshi Nakagawa. 2014. Formalizing word sampling for vocabulary prediction as graph-based active learning. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1374–1384, Doha, Qatar. Association for Computational Linguistics.

Sian Gooding and Manuel Tragut. 2022. One size does not fit all: The case for personalised word complexity models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 353–365, Seattle, United States. Association for Computational Linguistics.

Yue Guo, Joseph Chee Chang, Maria Antoniak, Erin Bransom, Trevor Cohen, Lucy Wang, and Tal August. 2024. Personalized jargon identification for enhanced interdisciplinary communication. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4535–4550, Mexico City, Mexico. Association for Computational Linguistics.

Yusuke Ide, Masato Mita, Adam Nohejl, Hiroki Ouchi, and Taro Watanabe. 2023. Japanese lexical complexity for non-native readers: A new dataset. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 477–487, Toronto, Canada. Association for Computational Linguistics.

Ming Ji and Jiawei Han. 2012. A variance minimization criterion to active learning on graphs. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 556–564, La Palma, Canary Islands. PMLR.

Ruili Jiang, Kehai Chen, Xuefeng Bai, Zhixuan He, Juntao Li, Muyun Yang, Tiejun Zhao, Liqiang Nie, and Min Zhang. 2024. A survey on human preference learning for large language models. *Preprint*, arXiv:2406.11191.

Qi Liu, Shiwei Tong, Chuanren Liu, Hongke Zhao, Enhong Chen, Haiping Ma, and Shijin Wang. 2019. Exploiting cognitive structure for adaptive learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 627–635, New York, NY, USA. Association for Computing Machinery.

Adam Nohejl, Akio Hayakawa, Yusuke Ide, and Taro Watanabe. 2024. Difficult for whom? A study of Japanese lexical complexity. In *Proceedings of the Third Workshop on Text Simplification, Accessibility and Readability (TSAR 2024)*, pages 69–81, Miami, Florida, USA. Association for Computational Linguistics.

OpenAI. 2025. GPT-4.1. `https://developers.openai.com/api/docs/models/gpt-4.1`.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Matthew Shardlow, Richard Evans, and Marcos Zampieri. 2022. Predicting lexical complexity in English texts: the Complex 2.0 dataset. *Language Resources and Evaluation*, 56(4):1153–1194.

Fabian M. Suchanek, Mehwish Alam, Thomas Bonald, Lihu Chen, Pierre-Henri Paris, and Jules Soria. 2024. YAGO 4.5: A large and clean knowledge base with a rich taxonomy. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, page 131–140, New York, NY, USA. Association for Computing Machinery.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 950–958, New York, NY, USA. Association for Computing Machinery.

Zhouhang Xie, Junda Wu, Yiran Jenny Shen, Raghav Jain, Yu Xia, Xintong Li, Aaron Chang, Ryan A. Rossi, Tong Yu, Sachin Kumar, Bodhisattwa Prasad Majumder, Jingbo Shang, Prithviraj Ammanabrolu, and Julian McAuley. 2025. A survey on personalized and pluralistic preference alignment in large language models. In *Second Conference on Language Modeling*.

Ikuya Yamada, Akari Asai, Jin Sakuma, Hiroyuki Shindo, Hideaki Takeda, Yoshiyasu Takefuji, and Yuji Matsumoto. 2020. Wikipedia2Vec: An efficient toolkit for learning and visualizing the embeddings of words and entities from Wikipedia. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 23–30, Online. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. 2003. Learning with local and global consistency. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'03, page 321–328, Cambridge, MA, USA. MIT Press.

Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University, Pittsburgh, PA.

Annotator	Acc.	Prec.	Rec.	F1
Annotator 1	0.7856	0.4842	0.7077	0.5750
Annotator 2	0.6791	0.5586	0.6542	0.6027
Annotator 3	0.6564	0.6498	0.7928	0.7142
Annotator 4	0.6948	0.5556	0.6786	0.6109
Annotator 5	0.6570	0.7629	0.7424	0.7525
Annotator 6	0.6878	0.6480	0.7740	0.7054
Annotator 7	0.6709	0.6464	0.7053	0.6746
Annotator 8	0.6286	0.6845	0.6838	0.6842
Average	*0.6825*	*0.6237*	*0.7173*	*0.6649*

Table 6: Per-annotator results on **Common Knowledge** articles.

Annotator	Acc.	Prec.	Rec.	F1
Annotator 1	0.7048	0.6778	0.6983	0.6879
Annotator 2	0.7351	0.7206	0.8657	0.7865
Annotator 3	0.7530	0.8324	0.8279	0.8301
Annotator 4	0.7410	0.7330	0.6953	0.7137
Annotator 5	0.7052	0.7228	0.7228	0.7228
Annotator 6	0.7844	0.8286	0.8580	0.8430
Annotator 7	0.7470	0.8098	0.7816	0.7955
Annotator 8	0.7809	0.8701	0.8481	0.8590
Avg (Chinese)	*0.7509*	*0.7899*	*0.8133*	*0.8003*
Avg (Japanese)	*0.7231*	*0.7279*	*0.7091*	*0.7183*
Average	*0.7439*	*0.7744*	*0.7872*	*0.7798*

Table 7: Per-annotator results on **Chinese Culture** articles. Chinese and Japanese indicate annotator nationalities.

A Per-Annotator Detailed Results

Tables 6–10 report the results of each annotator in each domain, computed over all articles of that domain. F1 varies considerably across annotators, for example from 0.575 to 0.753 on Common Knowledge. In the cultural domains, annotators whose background matches the domain tend to achieve higher F1 scores, consistent with Table 3.

Annotator	Acc.	Prec.	Rec.	F1
Annotator 1	0.7221	0.5034	0.8212	0.6242
Annotator 2	0.6865	0.7220	0.8090	0.7630
Annotator 3	0.7017	0.7226	0.7781	0.7493
Annotator 4	0.7527	0.7643	0.8586	0.8087
Annotator 5	0.8576	0.8911	0.9534	0.9212
Annotator 6	0.7152	0.6614	0.7404	0.6987
Annotator 7	0.6635	0.7152	0.6351	0.6728
Annotator 8	0.6682	0.7873	0.6786	0.7289
Avg (Chinese)	*0.6929*	*0.6853*	*0.7437*	*0.7062*
Avg (Japanese)	*0.8052*	*0.8277*	*0.9060*	*0.8650*
Average	*0.7210*	*0.7209*	*0.7843*	*0.7459*

Table 8: Per-annotator results on **Japanese Culture** articles. Chinese and Japanese indicate annotator nationalities.

Annotator	Acc.	Prec.	Rec.	F1
Annotator 1	0.8490	0.4359	0.5312	0.4789
Annotator 2	0.6265	0.7138	0.7205	0.7172
Annotator 3	0.6789	0.7240	0.7922	0.7566
Annotator 4	0.6082	0.5837	0.6383	0.6098
Annotator 5	0.7939	0.8956	0.8642	0.8796
Annotator 6	0.6327	0.6402	0.6194	0.6296
Annotator 7	0.6061	0.6172	0.6858	0.6497
Annotator 8	0.7122	0.7620	0.8649	0.8102
Avg (CS)	*0.6710*	*0.7161*	*0.7610*	*0.7372*
Avg (Semi.)	*0.7408*	*0.5380*	*0.5753*	*0.5543*
Average	*0.6884*	*0.6716*	*0.7146*	*0.6914*

Table 9: Per-annotator results on **Computer Science** articles. CS and Semi. indicate annotator fields.

Annotator	Acc.	Prec.	Rec.	F1
Annotator 1	0.7933	0.4200	0.3889	0.4038
Annotator 2	0.6087	0.6356	0.5034	0.5618
Annotator 3	0.6200	0.7232	0.6632	0.6919
Annotator 4	0.6533	0.4752	0.4848	0.4800
Annotator 5	0.7600	0.8843	0.8295	0.8560
Annotator 6	0.6833	0.7532	0.6705	0.7095
Annotator 7	0.5433	0.6629	0.6051	0.6327
Annotator 8	0.5900	0.7314	0.6275	0.6755
Avg (CS)	*0.6292*	*0.6854*	*0.6189*	*0.6496*
Avg (Semi.)	*0.7383*	*0.5866*	*0.5297*	*0.5567*
Average	*0.6565*	*0.6607*	*0.5966*	*0.6264*

Table 10: Per-annotator results on **Semiconductors** articles. CS and Semi. indicate annotator fields.

B Per-Domain Detailed Results

Table 11 reports per-domain results for all methods. All metrics use familiar as the positive class and are computed as in Table 2. Our method achieves the highest F1 on Chinese Culture (0.780 vs. 0.752 for *Qwen2.5-72B-Instruct-AWQ*) and the highest accuracy on Chinese Culture, Japanese Culture, and Computer Science, whereas *Qwen2.5-72B-Instruct-AWQ* achieves the highest F1 in the other four domains, together with the highest recall among non-trivial methods. The *All-Familiar* baseline attains perfect recall by labeling every entity as familiar, but at the cost of low precision. *GPT-4.1-mini* attains the highest precision in every domain but low recall, suggesting that in-context learning from 20 seed annotations under-predicts familiarity.

C Human Evaluation Details

Table 13 lists the complete set of all 90 Wikipedia pages used in our evaluation, organized by knowledge dimension and subcategory. We recruit eight annotators with diverse academic and cultural backgrounds (Table 12). Each participant annotates a deduplicated annotation list of Wikipedia entities extracted from the entire 90-page corpus, labeling each entity as either "Familiar" or "Unfamiliar". Annotations are performed offline via a structured spreadsheet containing the entity name, its Wikipedia abstract, and a binary annotation field. Participants are prohibited from consulting any external resources during the task and are instructed not to discuss their labels with others. No time limit is imposed. The complete annotation prompt is shown in Figure 3.

D Parameter Sensitivity Analyses

We examine the sensitivity of our framework to five hyperparameters, presented in pipeline order: the kNN expansion size k (§3.2), the Gaussian bandwidth σ (Equation 1), the Laplacian regularization γ (Equation 3), the annotation budget q (§3.3), and the propagation iterations T (§3.4). All sensitivity experiments use the default configuration (k=10, σ=0.75, γ=0.01, q=20, T=20, undirected propagation) while varying one parameter at a time.

kNN Neighbors k. We vary $k \in \{0, 3, 5, 10, 15, 20, 30\}$. Performance is remarkably stable (F1: 0.6990–0.7018), with less than 0.3 percentage points of variation. Even k=0 (no expansion) performs comparably (F1: 0.7004), indicating that the fully connected core graph already provides sufficient propagation pathways. We adopt k=10 as the default.

Gaussian Bandwidth σ. We vary $\sigma \in \{0.25, 0.50, 0.75, 1.00, 1.50\}$. The system is robust across $\sigma \in [0.50, 1.50]$ (F1: 0.7017–0.7081), with only σ=0.25 showing a notable drop (F1: 0.6613) due to overly concentrated edge weights that suppress recall. We adopt σ=0.75 as the default.

Laplacian Regularization γ. We vary $\gamma \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1.0\}$. The system is highly robust, with F1 varying by less than 0.2 percentage points as γ ranges from 0.001 to 1.0. We adopt γ=0.01 as the default.

Annotation Budget q. We vary $q \in \{10, 15, 20, 25, 30\}$. Performance improves steadily from q=10 (F1: 0.6739) to q=25 (F1: 0.7051), then plateaus. We adopt q=20 as a practical default balancing annotation effort and prediction quality.

Propagation Iterations T. We vary $T \in \{1, 3, 5, 10, 15, 20, 30, 50, 100\}$. Propagation converges rapidly: T=1 already achieves F1 of 0.7020, and scores stabilize by T=15. Total variation is less than 0.1 percentage points. We set T=20 as a conservative default.

E Models and Computational Budget

Our method trains no model. It uses the ReFinED entity linker (Ayoola et al., 2022) (about 154M parameters) and pre-trained 100-dimensional English Wikipedia2Vec embeddings (Yamada et al., 2020). Entity linking for the 90 articles takes about 12 minutes on a single NVIDIA V100 GPU. All remaining steps run on CPU and finish for all annotators and domains in about one minute (4 CPU cores, 16 GB of memory).

GPT-4.1-mini is accessed through the OpenAI API, and its size has not been disclosed. *Qwen2.5-72B-Instruct-AWQ* (72.7B parameters, 4-bit AWQ quantization) is served with vLLM on two NVIDIA RTX 6000 Ada Generation GPUs and takes about 6 hours for all annotators and articles (about 12 GPU hours).

Domain	Method	Acc	Prec	Rec	F1
Common Knowledge	*All-Familiar*	0.466	0.466	1.000	0.622
	GPT-4.1-mini	0.646	**0.693**	0.344	0.458
	Qwen2.5-72B-Instruct-AWQ	**0.690**	0.610	**0.812**	**0.692**
	Ours	0.683	0.624	0.717	0.665
Chinese Culture	*All-Familiar*	0.606	0.606	1.000	0.748
	GPT-4.1-mini	0.641	**0.823**	0.488	0.610
	Qwen2.5-72B-Instruct-AWQ	0.692	0.713	**0.801**	0.752
	Ours	**0.744**	0.774	0.787	**0.780**
Japanese Culture	*All-Familiar*	0.576	0.576	1.000	0.717
	GPT-4.1-mini	0.702	**0.781**	0.621	0.687
	Qwen2.5-72B-Instruct-AWQ	0.689	0.692	**0.834**	**0.749**
	Ours	**0.721**	0.721	0.784	0.746
Computer Science	*All-Familiar*	0.564	0.564	1.000	0.697
	GPT-4.1-mini	0.674	**0.766**	0.469	0.574
	Qwen2.5-72B-Instruct-AWQ	0.679	0.654	**0.776**	**0.699**
	Ours	**0.688**	0.672	0.715	0.691
Semiconductors	*All-Familiar*	0.552	0.552	1.000	0.688
	GPT-4.1-mini	0.643	**0.825**	0.376	0.513
	Qwen2.5-72B-Instruct-AWQ	**0.722**	0.676	**0.813**	**0.731**
	Ours	0.657	0.661	0.597	0.626

Table 11: Per-domain results for all methods. Bold marks the best result per metric among methods other than *All-Familiar*.

Annotation Prompt for Entity Familiarity

`# Task`
You will be presented with a list of Wikipedia entities extracted from an article corpus. For each entity, judge whether you are **familiar** or **unfamiliar** with it. This task maps your unique knowledge boundary. It is **not** a test of intelligence.

`# Definitions`
Familiar: You have encountered the entity before and can provide a brief, correct description (even if vague). General understanding is sufficient.
Unfamiliar: You have never heard of the entity, or you recognize the name but cannot describe what it refers to.

`# Guidelines`
1. Judge based **solely on current knowledge**. Do not look up any entity.
2. There are no right or wrong answers. We seek your genuine knowledge profile.
3. If uncertain, lean towards "Unfamiliar" for a conservative boundary estimate.
4. Read the provided Wikipedia abstract to assist recognition before judging.

Figure 3: Annotation prompt provided to all participants. Instructions were presented in English only.

Annotator	Expertise	Nationality
Annotator 1	Semiconductor	Chinese
Annotator 2	Computer Science	Chinese
Annotator 3	Computer Science	Chinese
Annotator 4	Computer Science	Japanese
Annotator 5	Computer Science	Japanese
Annotator 6	Semiconductor	Chinese
Annotator 7	Computer Science	Chinese
Annotator 8	Computer Science	Chinese

Table 12: Participant demographics.

Category	Wikipedia Pages
Common Knowledge (30)	Coffee, Tea, Soap, Pencil, Mirror, Umbrella, Honey, Map, Clock, Candle, Glass, Bicycle, Salt, Bread, Plastic, Vaccine, Leather, Camera, Magnet, Concrete, Cotton, Silk, Microscope, Refrigerator, Elevator, Calendar, Wheel, Thermometer, Natural Rubber, Brick
Computer Science (15)	Algorithm, Machine Learning, Virtual Machine, Cryptography, Operating System, Software Engineering, Database, Compiler, Computer Network, Neural Network, Artificial Intelligence, World Wide Web, Bluetooth, Computer Virus, Open-source Software
Semiconductors (15)	Semiconductor, Integrated Circuit, Photolithography, Wafer, Transistor, MOSFET, Light-emitting Diode, Doping, Flash Memory, Silicon, Semiconductor Device Fabrication, CMOS, Dynamic Random-access Memory, Electron Mobility, Chemical Vapor Deposition
Chinese Culture (15)	Chinese New Year, Mid-Autumn Festival, Mahjong, Hot Pot, Bubble Tea, Peking Opera, Calligraphy, Dim Sum, Qipao, Chinese Tea Culture, Chinese Martial Arts, Tai Chi, Chinese Dragon, Feng Shui, Chopsticks
Japanese Culture (15)	Sushi, Anime, Onsen, Shinkansen, Kimono, Matcha, Ramen, Sake, Sumo, Kabuki, Origami, Manga, Geisha, Zen, Ikebana

Table 13: Complete list of 90 Wikipedia pages used in the evaluation, organized by knowledge dimension.

Metric	k=0	k=3	k=5	k=10	k=15	k=20	k=30
Accuracy	0.6976	0.6981	0.6977	0.6985	0.6973	0.6967	0.6964
Precision	0.6902	0.6901	0.6891	0.6903	0.6895	0.6862	0.6888
Recall	0.7176	0.7200	0.7201	0.7200	0.7177	0.7183	0.7165
F1	0.7004	0.7018	0.7014	0.7017	0.7003	0.6990	0.6991

Table 14: Sensitivity to kNN neighbors k.

Metric	σ=0.25	σ=0.50	σ=0.75	σ=1.00	σ=1.50
Accuracy	0.6723	0.6958	0.6985	0.7007	0.7047
Precision	0.7036	0.6922	0.6903	0.6914	0.6923
Recall	0.6286	0.7220	0.7200	0.7224	0.7295
F1	0.6613	0.7036	0.7017	0.7037	0.7081

Table 15: Sensitivity to Gaussian bandwidth σ.

Metric	γ=0.001	γ=0.005	γ=0.01	γ=0.05	γ=0.1	γ=0.5	γ=1.0
Accuracy	0.6974	0.6969	0.6985	0.6988	0.6974	0.6974	0.6973
Precision	0.6885	0.6885	0.6903	0.6885	0.6878	0.6899	0.6879
Recall	0.7205	0.7184	0.7200	0.7233	0.7212	0.7195	0.7210
F1	0.7010	0.7002	0.7017	0.7021	0.7009	0.7011	0.7008

Table 16: Sensitivity to Laplacian regularization γ.

Metric	q=10	q=15	q=20	q=25	q=30
Accuracy	0.6765	0.6902	0.6985	0.7014	0.7016
Precision	0.6791	0.6872	0.6903	0.6916	0.6876
Recall	0.6766	0.7000	0.7200	0.7249	0.7255
F1	0.6739	0.6908	0.7017	0.7051	0.7034

Table 17: Sensitivity to annotation budget q.

Metric	T=1	T=3	T=5	T=10	T=15	T=20	T=30	T=50	T=100
Accuracy	0.7003	0.6980	0.6983	0.6983	0.6985	0.6985	0.6985	0.6985	0.6984
Precision	0.6943	0.6895	0.6901	0.6901	0.6903	0.6903	0.6903	0.6903	0.6902
Recall	0.7157	0.7197	0.7198	0.7200	0.7200	0.7200	0.7200	0.7203	0.7203
F1	0.7020	0.7012	0.7015	0.7016	0.7017	0.7017	0.7017	0.7018	0.7018

Table 18: Sensitivity to propagation iterations T.

F LLM Baseline Prompts

Both *GPT-4.1-mini* and *Qwen2.5-72B-Instruct-AWQ* receive identical prompts constructed from the same 20 seed entities selected by V-optimality. Each unlabeled entity is accompanied by the first sentence of its Wikipedia abstract to disambiguate the concept. Further details of the baseline setup are given in §4.1.

System Prompt (LLM Baselines)

```
You are an expert at predicting a reader's
vocabulary knowledge. Given a set of
concepts that a reader has marked as
familiar or unfamiliar, predict whether the
reader would be familiar with each of the
remaining concepts. Consider the reader's
apparent expertise level, domain knowledge,
and the semantic relationships between
concepts. Each concept is accompanied by
a brief description to help you understand
what it refers to.
```

User Prompt (LLM Baselines)

```
A reader was asked about some concepts and
gave these answers:
FAMILIAR (known): {known_entities}
UNFAMILIAR (unknown): {unknown_entities}
Based on this pattern, predict whether the
reader would be familiar with each of the
following concepts. For EACH concept,
respond with exactly one line in the
format:
  concept_name: familiar
or
  concept_name: unfamiliar
Do NOT add any other text, explanation, or
numbering. Output ONLY the lines in the
specified format.
Concepts to predict (with brief
descriptions):
- {entity_name}: {description}
- ...
```

G YAGO Predicate Selection

When constructing the YAGO adjacency signal a_{ij} described in §5, we use a manually selected set of YAGO 4.5 relation types (predicates). Table 19 lists them, grouped by their role in connecting semantically related entities.

Category	Predicate URI
Taxonomic	`rdfs:subClassOf`, `rdf:type`
Mereological	`schema:isPartOf`, `schema:hasPart`
Associative	`schema:genre`, `schema:knowsAbout`
Equivalent	`owl:sameAs`, `skos:broader`

Table 19: YAGO relation types (predicates) used to construct the adjacency signal a_{ij} (Equation 9).

H Use of AI Assistants

We employed AI-based language assistants as supportive tools during the preparation of this manuscript. Specifically, AI was used to assist in drafting, refining, and polishing the English expressions to ensure clarity and coherence. Furthermore, AI tools provided assistance in developing data-processing and experiment code scripts. It is important to emphasize that all scientific content, experimental designs, and methodological frameworks were independently conceived and verified by the authors. AI assistants were not involved in conducting human evaluation annotations; all such data collection was performed solely by human participants without AI intervention.

I Artifact Licenses

All external artifacts used in this work are listed below along with their licenses and intended use.

- **ReFinED** (Ayoola et al., 2022): Released under the Apache 2.0 license. Used for entity linking to extract Wikipedia entities from documents.
- **Wikipedia2Vec** (Yamada et al., 2020): Released under the Apache 2.0 license. Used to obtain pre-trained entity embeddings for graph construction.
- **YAGO 4.5** (Suchanek et al., 2024): Released under the CC BY-SA 3.0 license. Used to enrich the entity subgraph with structured relational knowledge.
- **GPT-4.1-mini** (OpenAI, 2025): Accessed via the OpenAI API under OpenAI's terms of service. Used as an LLM baseline for entity familiarity prediction.
- **Qwen2.5-72B-Instruct-AWQ** (Qwen et al., 2025): Released under the Qwen License. Used as an LLM baseline for entity familiarity prediction.

All artifacts are used for non-commercial research purposes consistent with their respective licenses and intended use.