DynamicER: Resolving Emerging Mentions to Dynamic Entities for RAG

Jinyoung Kim **Dayoon Ko** **Gunhee Kim**
Seoul National University
jiny1623@snu.ac.kr dayoon.ko@vision.snu.ac.kr gunhee.kim@snu.ac.kr
https://github.com/jiny1623/DynamicER

Abstract

In the rapidly evolving landscape of language, resolving new linguistic expressions in continuously updating knowledge bases remains a formidable challenge. This challenge becomes critical in retrieval-augmented generation (RAG) with knowledge bases, as emerging expressions hinder the retrieval of relevant documents, leading to generator hallucinations. To address this issue, we introduce a novel task aimed at resolving emerging mentions to dynamic entities and present DYNAMICER benchmark. Our benchmark includes dynamic entity mention resolution and entity-centric knowledge-intensive QA task, evaluating entity linking and RAG model's adaptability to new expressions, respectively. We discovered that current entity linking models struggle to link these new expressions to entities. Therefore, we propose a temporal segmented clustering method with continual adaptation, effectively managing the temporal dynamics of evolving entities and emerging mentions. Extensive experiments demonstrate that our method outperforms existing baselines, enhancing RAG model performance on QA task with resolved mentions.

1 Introduction

In the real world, large amounts of textual information are constantly being generated at an incredible rate. The dynamic nature of human language, characterized by the continuous emergence of new expressions, presents multiple significant challenges (Hirschberg and Manning, 2015). The way we refer to named entities changes over time, influenced by the shifts that include the use of metaphors, adoption of slang, creation of euphemisms, and other linguistic evolutions (Li et al., 2020). For example, consider the named entity "Elon Musk". Over time, various mentions are used to refer to him, such as "the Tesla CEO", or "the tech billionaire". Emerging slang or metaphoric expressions like "real-life Iron Man" or "Mars man" also appear. As a dynamic entity, his attributes change over time as he was initially known as "PayPal co-founder", later as "Hyperloop visionary", and more recently as "Twitter owner". All of these expressions refer to the same entity at different time points, yet a system must be able to recognize them despite the dynamic linguistic landscape. Therefore, it is crucial for a system to resolve these new expressions, accurately linking them to evolving entities in a continuously updating knowledge base (KB).

Figure 1: Motivation of our DYNAMICER benchmark. New mentions referring to the same entity are constantly created over time: as Shohei Ohtani transfers from the LA Angels to the LA Dodgers, he is referred to by new mentions such as 'The Dodgers' number 17.' We contribute a dynamic entity resolution dataset, along with two benchmark tests: traditional entity linking and entity-centric question-answering in the RAG context.

Recently, KBs have been critically utilized in the retrieval-augmented generation (RAG) framework (Lewis et al., 2020; Guu et al., 2020; Izacard et al., 2023; Asai et al., 2023), where a retriever fetches relevant documents from KBs, allowing a generator like large language models (LLMs) to provide accurate answers. As LLMs become capable of learning with prompt, studies have utilized RAG for knowledge-intensive open-domain tasks. Following this trend, efforts (Liska et al., 2022; Dhingra et al., 2022; Neelam et al., 2022) have been made to manage dynamically evolving knowledge with RAG. In particular, some studies (Kasai et al., 2024; Ko et al., 2024) have shown that in time-sensitive knowledge-intensive benchmarks, LLMs inevitably produce outdated answers if retrieval fails. One significant reason for retrieval failure is the emergence of new mentions, which prevents the retriever from functioning properly (Sciavolino et al., 2021; Mallen et al., 2023). However, to the best of our knowledge, no approach has tackled the challenge of resolving new expressions of named entities evolving over time for the RAG framework.

To overcome this limitation, we call for a dynamic entity resolution task that links emerging mentions to dynamic entities. We contribute the DYNAMICER (**Dynamic** **E**ntity **R**esolution for Emerging Mentions) benchmark, designed to study the resolution of continuously evolving entities and their newly appearing mentions. As illustrated in Figure 1, DYNAMICER annotates emerging mentions found in social media documents, linking these mentions to corresponding named entities within a KB. DYNAMICER is structured as a sequence of time segments to evaluate models resolving mentions that are not recognized in earlier time steps. We also introduce a dynamic entity-centric question-answering (QA) task where named entities in question are substituted into emerging mentions over time. This QA task examines the impact of emerging mentions on the retriever's accuracy and aims to evaluate the end-to-end performance of the RAG methods. Table 1 presents a comparison of DYNAMICER with other benchmarks.

When a new mention first appears, it may be challenging to resolve it using only a single document since it is likely to have high lexical variation and insufficient context. Although prior works have used mentions in multiple documents collectively (Ganea and Hofmann, 2017; Le and Titov, 2018; Angell et al., 2021; Agarwal et al., 2022), there is a potential risk when jointly handling mentions from different time steps. The definition or attributes of entities may evolve across time steps, and neighbor mentions referenced together also change accordingly. To address this issue, we propose a method that continuously clusters entities and a set of mentions at each time step and updates the evolving entity cluster representations.

Our contributions are summarized as follows:

1. We introduce DYNAMICER as the first benchmark for linking emerging mentions to dynamic entities, measuring the capability of RAG models to resolve and adapt to newly emerging expressions.

2. We propose a temporal segmented clustering with continual adaptation, which considers temporal dynamics to distinguish between entities and mentions more effectively, especially when new mentions emerge.

3. We empirically show that our method outperforms other entity linking methods and that resolving new mentions is beneficial for RAG performance in QA tasks.

2 Related Work

Entity Linking. Entity linking (Hoffart et al., 2011; Guo and Barbosa, 2018) aims to match an entity mention to a unique named entity in a KB such as Wikipedia pages. There has been much research on how to correctly link varied mentions of the same entity. For instance, Andy et al. (2017) design an algorithm to identify entities from social media during the 3-4 hour Grammy Awards, constructing an alias list for short-term use. Botzer et al. (2021) collect a Reddit entity linking dataset, demonstrating that models trained on conventional text encounter difficulties with the unique formats and lexical variations prevalent in social media. In the biomedical domain, Mohan and Li (2018) propose the MedMentions dataset, which compiles a comprehensive biomedical corpus with entity mention annotations. However, the static nature of this domain fails to accommodate the time-evolving linguistic evolution. The zero-shot setting of entity linking (Lin et al., 2017; Logeswaran et al., 2019) targets linking entities unseen in training time. This task focuses on domain adaptation to resolve new entities, rather than on handling the mention variation of specific entities over time.

	MEDMENTIONS (2018)	ZERO-SHOT EL (2019)	REDDIT EL (2021)	TEMPEL (2022)	DYNAMICER (Ours)
Source	PubMed	Wikias	Social media	Wikipedia	Social media
Domain	Biomedical	Fictional universes	General	General	Sports
Size	350K (4K Docs)	70K (16 Wikias)	17K (619 Posts)	240K	70K (20K Docs)
Temporal dynamics	✗	✗	✗	✓	✓
Mention variations	✓	✓	✓	✗	✓
Tasks	Entity linking	Entity linking	Entity linking	Continuous entity linking	Dynamic entity mention resolution & Entity-centric QA

Table 1: Comparison of our DYNAMICER with existing entity linking benchmarks. The source indicates from which the dataset is collected. The domain shows the type of data content. The size displays the number of mentions along with the number of documents, Wikias, or posts. The temporal dynamics represent whether the dataset evolves or changes over time. The mention variations show whether the dataset is annotated with alias lists of mentions.

For temporal entity linking, the TempEL dataset offers detailed tracking and annotation of changes in existing entities as well as the emergence of new entities across multiple temporal snapshots. Our annotation is similar to TempEL in that we handle temporal development of existing entities. However, our dataset also considers the continuous emergence of new mentions of the same entity, highlighting that the way it is referenced varies across multiple points in time.

Coreference Resolution. Coreference resolution (CR) (Pradhan et al., 2012; Webster et al., 2018) evaluates a model's ability to match entities with their antecedents. The task is to group all spans that point to the same objects in a context by detecting mentions. Lee et al. (2017) integrates these two processes in an end-to-end manner by considering all spans as potential coreference candidates and learning a conditional probability distribution for clustering. Joshi et al. (2020) extends BERT by training via masked contiguous random spans and predicting the spans using boundary representation.

Cross-document CR (CDCR) (Cybulska and Vossen, 2014; Webster et al., 2018) resolves coreference across multiple documents. Caciularu et al. (2021) utilizes long-range transformers to encode multiple related documents. Allaway et al. (2021) sequentially adds each mention to cluster candidates while incrementally updating the coreference candidate cluster representation. Our dynamic entity linking seems similar to CDCR in that both link the mentions referring to the same entity over documents. However, our task is different since it aims to link varied mentions to the evolving entities of a continuously updating KB.

RAG with Dynamic Corpus. Since continuously retraining LLMs with up-to-date data is demanding, RAG has been employed to handle temporal adaptability. For instance, Liska et al. (2022) and Kasai et al. (2024) respectively propose dynamic QA tasks with time-stamped and newly published news articles, demonstrating that retrieving up-to-date documents can improve generation results. Moreover, Dhingra et al. (2022) and Margatina et al. (2023) introduce a cloze query to evaluate the acquisition of temporal knowledge. Neelam et al. (2022) proposes Knowledge Base QA (KBQA) tasks to evaluate the ability of temporal reasoning. Recently, Ko et al. (2024) introduces dynamically evolving open-domain QA and dialogue benchmarks along with a novel training-free retrieval-interactive LLM framework. While existing works focus on the temporal adaptability of models for retrieval and reasoning, they do not specifically address the dynamic nature of entity expressions, which is crucial for applications requiring precise entity linking over time.

3 The DYNAMICER Dataset

DYNAMICER consists of two tasks: an entity linking task and an entity-centric QA task. The entity linking task focuses on resolving emerging expressions that appear over time to entities in a KB, while the entity-centric QA task evaluates RAG models in answering entity-specific questions. We construct DYNAMICER through a pipeline, whose key idea at each stage is to first generate automatically using LLMs, and then thoroughly verify the quality with human review: (1) selecting and fil-

tering textual corpora (§ 3.1), (2) identifying mentions of target entities (§ 3.2), (3) annotating the appropriate entity for each mention (§ 3.3), and (4) generating QA pairs using resolved entities (§ 3.4). The prompt templates for dataset generation are provided in Appendix F.

3.1 Corpora Collection

Post Selection. We choose the sports domain to capture the mention variations of famous athletes, teams, and coaches, given its inherently dynamic nature. This domain is particularly suitable since it features diverse naming conventions, such as nicknames and abbreviations for entities. Moreover, frequent updates and news about events like matches and player transfers contribute to its dynamic nature. The sports domain is also event-driven with clear temporal markers like seasons and tournaments.

We target soccer and baseball for corpora selection. Specifically, for soccer, we select the top 15 teams according to *Forbes World's Most Valuable Soccer Teams*[1] and leagues to which these teams belong. For baseball, we select 30 teams from Major League Baseball. Using the `/tagged` method in Tumblr API, we download all posts tagged with our selected hashtags, focusing on posts from 2023-05-01 to 2024-04-30. The full list of hashtags can be found in Appendix C.

Initial Filtering. We filter the posts with fewer than 50 or more than 3000 characters to exclude content that is either too brief to provide sufficient contextual meaning or too lengthy to potentially divert attention with extraneous information. Additionally, we use the FastText module (Joulin et al., 2016a,b) to ensure text is written in English, filtering out the posts that are not confidently identified as English.

3.2 Mention Identification

We first identify expressions referring to named entities using the GPT-4 turbo (Achiam et al., 2023). The prompt we use is as follows: '*Please identify all expressions in the given text that explicitly name or describe a player, coach, or team. This includes direct names, nicknames, and any role-specific references (like positions or accolades) that refer to a particular individual or team.*' We refrain from using typical named entity recognition (NER) models since they struggle to identify long expressions, such as "*the defending National League champion*". We ignore posts for which GPT-4 detects fewer than two expressions, as they lack sufficient contextual information for a resolution dataset.

[1] https://www.forbes.com/lists/soccer-valuations/

	0506	0708	0910	1112	0102	0304
Soccer						
Documents	2346	2658	2803	2560	3081	2143
Mentions	8255	8817	9746	9031	10284	7541
Unique expressions	2786	3202	3231	2960	3237	2694
Emerging expressions	-	2320	1947	1525	1636	1148
QA pairs	1015	945	888	603	616	444
Baseball						
Documents	672	894	822	290	448	984
Mentions	1725	3279	3254	903	1073	3501
Unique expressions	813	1409	1306	529	747	1827
Emerging expressions	-	1071	848	255	375	980
QA pairs	734	911	730	194	320	800

Table 2: Statistics of DYNAMICER. Each column represents a two-month period within the dataset. Unique expressions denote distinct surface forms in each month, while emerging expressions denote distinct surface forms that appear for the first time in each month. Refer to Figure 3 for the number of distinct mentions for each of the Top-50 entities.

3.3 Entity Annotation

Once expressions are mined by GPT-4, we use a substring matcher to highlight each expression in order, followed by human verification. We employ a dedicated team of workers to annotate the data. Please refer to Ethics Statement for the details. We instruct the annotators to adjust the offset of the expression if the highlighting is incorrect and to additionally highlight any missing expressions that refer to players, coaches, or teams. Next, the annotators link each expression to the corresponding Wikipedia entity. They search Wikipedia for a suitable entity and submit a valid URL of the Wikipedia page to the system. If a valid Wikipedia entity cannot be found, or if the context from the post is too ambiguous to resolve the expressions, annotators label it as NOT VALID, which is then further filtered out.

3.4 Entity-Centric QA Pairs

Based on the previous annotation, we create entity-centric QA pairs, which require entity resolution to provide accurate answers. Our basic idea is to replace each entity name in the questions with its

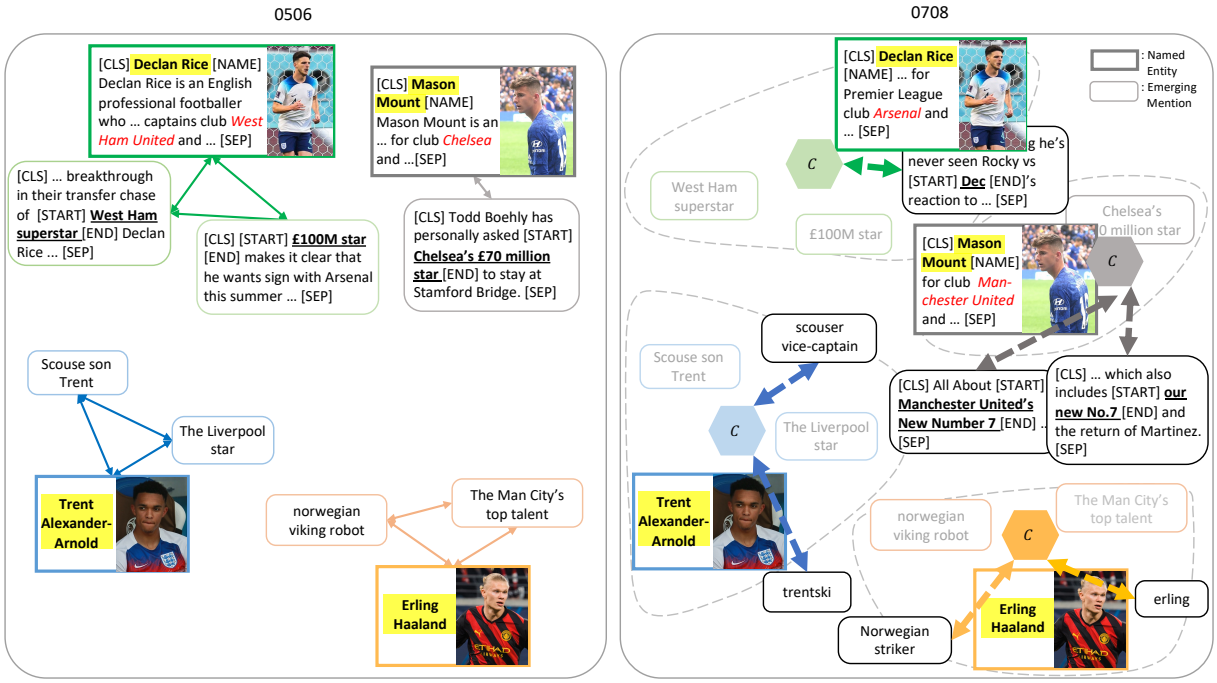

Figure 2: An illustrative example of TempCCA. C denotes the representation of entity clusters formed in the previous time step. The rectangular boxes contain the entity input tokens, and the rounded boxes contain the input tokens for mention context. Entity names are highlighted, and mentions are underlined. TempCCA uses resolved mentions from the previous time step to form clusters, utilizing these cluster representations to resolve mentions in the subsequent time step. The attributes of entities that have changed are depicted in red text.

various mentions. However, this can introduce ambiguity, as some mentions may not clearly identify the entity without additional context. For example, mentions like *The Bronx Bombers* are unambiguous and can be identified as NEW YORK YANKEES without context, whereas mentions like *the winning team* are not explicit without context. Hence, before creating QA pairs, we filter out ambiguous cases that are checked by the prompt to GPT-4: '*Select the mentions from the list that unambiguously refer to {entity} without context.*'. We further perform human verification for the remaining mentions.

Finally, we use the Wikipedia description of each entity to generate the knowledge-intensive QA pairs. To evaluate temporal challenges, we use Wikipedia articles from the revision that corresponds to the time when the mentions first appeared. With the description, we prompt GPT-4: '*Below is the description of {entity}. Please generate a question-answer pair regarding {entity}. The entity name itself should be included.*'. We instruct GPT-4 to enclose the entity within the bracket in the question text. Once QA pairs are generated, we replace the bracketed entity name in each question with its varied mention. The generated QA pairs are then subjected to human validation to ensure the accuracy of the question-answer pair. The details of generating QA pairs are provided in Appendix D. Finally, Table 2 shows the statistics of the labeled dataset.

4 Approach

Resolving new expressions based solely on the document where they first appear can be challenging due to their low lexical similarity to the entity name and the ambiguity of context. Thus, it can be advantageous to consider multiple documents that share similar contexts and expressions to resolve these mentions jointly. Previous works have studied this joint clustering approach (Ganea and Hofmann, 2017; Le and Titov, 2018; Angell et al., 2021; Agarwal et al., 2022), but they assume a static scenario and resolve the mentions without considering the time dimension. In our problem, on the other hand, entities evolve over time, with changes in definitions or attributes such as status, role, affiliation, or characteristics. As exemplified in Figure 1, it is hard to find the coreference between "*The Angels' superstar*" and "*The Dodgers' number 17*" for Shohei Ohtani. Instead, clustering mentions that appear at similar time steps could

be feasible since they share events or contexts at similar time steps. Therefore, we propose a temporal segmented clustering approach with continuous adaptation (TempCCA), as shown in Figure 2. Our approach follows the joint clustering methods as prior works but continuously clusters the emerging mentions at each time step and further utilizes the cluster representation to resolve mentions in the next time step.

4.1 Dual Encoder Clustering

We follow the dual encoder clustering approach from Agarwal et al. (2022). We construct a weighted graph G where the nodes represent the combined set of entities $\mathcal{E}$ and mentions $\mathcal{M}$. We then cluster these nodes based on the affinity between each pair of nodes. The weight of each edge is defined by affinity functions, ϕ and ψ; the former measures affinity between an entity and a mention and the latter is between mentions. For $e \in \mathcal{E}$ and $m_i, m_j \in \mathcal{M}$, we define the weight $w_{e,m_i} = -\phi(e, m_i)$ and $w_{m_i,m_j} = -\psi(m_i, m_j)$. Each affinity function is formulated by the inner product of corresponding node embeddings:

$$\phi(e, m_i) = \mathbf{u}_{\mathrm{C}}(e)^\top \mathbf{u}_{\mathrm{M}}(m_i), \quad (1)$$

$$\psi(m_i, m_j) = \mathbf{u}_{\mathrm{M}}(m_i)^\top \mathbf{u}_{\mathrm{M}}(m_j) \quad (2)$$

where $\mathbf{u}_{\mathrm{C}}(e)$ denotes the embedding of entity cluster formulated in the last previous time step, and $\mathbf{u}_{\mathrm{M}}(m_i)$ denotes the mention representation. The difference between Agarwal et al. (2022) and our work is that we formulate an entity cluster embedding, rather than using the pure output from the entity encoder.

For the entity encoder, the input tokens are structured as follows: `[CLS]` e_n `[NAME]` e_d `[SEP]`, where e_n is the name of entity, and e_d is the description of entity. We adopt a special token `[NAME]` to separate the name and description of the entity. For the mention encoder, the input tokens are structured as follows: `[CLS]` c_l `[START]` m_i `[END]` c_r `[SEP]`, where c_l, c_r refer to the context to the left and right of the mention m_i within the document. We adopt special tokens `[START]`, and `[END]` to indicate the mention span. The mention representation is defined by the output of the mention encoder, regardless of the time step.

4.2 Continuous Training

At the initial time step, each entity forms a single cluster. The representation of a single cluster is simply defined by the output of the entity encoder. Using the obtained representations at the initial time step, we train the affinity function and their affiliated encoders. We adopt an arborescence-based clustering approach (Agarwal et al., 2022). Training objectives, including positive and negative sampling, are shown in Appendix E.

At each subsequent time step, we utilize the most recent previously resolved mentions to form an entity cluster representation:

$$\mathbf{u}_{\mathrm{C}}(e) = \alpha\, \mathbf{Enc}_{\mathrm{E}}(e) + (1-\alpha)\, \frac{1}{|\mathcal{C}(e)|} \sum_{m_i \in \mathcal{C}(e)} \mathbf{Enc}_{\mathrm{M}}(m_i),$$

where $\mathbf{Enc}_{\mathrm{E}}(e)$ ($\mathbf{Enc}_{\mathrm{M}}(m_i)$) denotes the output of the entity (mention) encoder for the input token from entity e (mention m_i). $\mathcal{C}(e)$ represents mentions linked to entity e in the previous time step. If the previous time step is the training phase, we use gold linking. If it is the test phase, we formulate $\mathcal{C}(e)$ with predicted linking. The hyperparameter α is set to optimize the affinity models.

5 Experiments

We investigate the following research questions.

1. How well does our method resolve emerging mentions compared to existing entity linking and coreference methods?
2. Can resolving new mentions assist the RAG model in a knowledge-intensive task?
3. In which cases does this resolution contribute to its generative capabilities?

5.1 Experimental Setup

5.1.1 Entity Linking Task

Baselines. We use the following models as baselines: (i) **SpEL** (Shavarani and Sarkar, 2023): the structured prediction entity linking approach, achieving the state-of-the-art performance on the AIDA-CoNLL Dataset (Hoffart et al., 2011), (ii) **c-SpEL**: continuously trained SpEL over each time segment, (iii) **ArboEL** (Agarwal et al., 2022): the state-of-the-art model on MedMentions (Mohan and Li, 2018), and (iv) **TempCCA**: our temporal segment clustering approach with continuous adaptation. We use the dual-encoder setting from ArboEL for fair evaluation.

Method	Set 1 (0.0 - 0.2)			Set 2 (0.2 - 0.4)			Set 3 (0.4 - 0.6)			Set 4 (0.6 - 0.8)			Set 5 (0.8 - 1.0)			Total		
	1112	0102	0304	1112	0102	0304	1112	0102	0304	1112	0102	0304	1112	0102	0304	1112	0102	0304
SpEL	56.72	32.69	32.65	60.34	55.67	58.82	63.57	62.68	64.58	78.66	81.12	81.02	78.80	77.57	76.28	73.28	72.10	70.73
c-SpEL	59.70	31.73	33.67	58.65	56.20	59.05	63.57	60.35	65.05	78.17	80.56	80.08	80.41	75.69	75.03	73.60	70.97	70.24
ArboEL	60.33	50.89	54.95	78.18	73.16	72.55	84.78	80.21	82.59	**91.36**	**90.11**	90.52	92.89	91.66	95.02	87.67	84.67	86.03
TempCCA (Ours)	**68.60**	**52.31**	**58.42**	**80.00**	**75.03**	**75.34**	**85.73**	**83.30**	**83.61**	90.96	90.00	**90.62**	**94.73**	**93.19**	**95.51**	**88.96**	**86.19**	**87.00**

Table 3: Results of the entity linking task by lexical similarity and time segment.

	Set 1	Set 2	Set 3	Set 4	Set 5	Total
1112	242	1430	1829	2191	3339	9031
0102	281	1658	2246	2559	3540	10284
0304	202	1265	1769	2058	2247	7541

Table 4: The number of mentions for each bin of lexical similarity

Training and Inference. We divide our entity linking dataset into disjoint training and test time steps in timeline; specifically, documents dated from 2023-05-01 to 2023-10-31 constitute the training time steps, while documents from 2023-11-01 to 2024-04-30 form the test time steps. For each training time step, we further split the data into training and validation sets. For soccer, across the time steps 0506, 0708, and 0910, we use 6681, 7042, and 7783 mentions for training, and 1574, 1775, and 1963 mentions for validation, respectively. For baseball, we use 1417, 2648, and 2592 mentions for training, and 308, 631, 662 mentions for validation, respectively.

TempCCA and c-SpEL undergo continuous training and inference. Specifically, we divide the documents into two-month intervals, resulting in three cycles (0506, 0708, 0910) of continuous training and three cycles (1112, 0102, 0304) of continuous inference.

5.1.2 Entity-Centric QA

Baselines. We use four types of baselines for the entity-centric QA: (i) **LLM** (e.g., Llama3-8B-Instruct), (ii) **LLM-ER**: LLM with the top-1 entity linking prediction, (iii) **RaLM** (Ram et al., 2023): LLM with concatenated top-k retrievals, (iv) **RaLM-CoT**: RaLM with a prompt similar to zero-shot Chain-of-Thought (Kojima et al., 2022), where we first ask the LLM to resolve the mention in the question to an entity and then answer the question, and (v) **RaLM-ER**: RaLM with the top-1 entity linking prediction. We use the E5 (Wang et al., 2022) as the retriever and Llama3-8B-Instruct as the generator (see Llama3 Documentation), which are state-of-the-art models.

For LLM-ER and RaLM-ER, we utilize TempCCA to perform entity linking to resolve target mentions in question and then provide the top-1 entity prediction in the LLM's prompt. To provide TempCCA's top-1 entity linking prediction in the prompt for LLM-ER and RaLM-ER, we insert a sentence '*The {mention} may also be referred to as {top-1 entity prediction}.*' right before the question. Additionally, we provide the LLM with top-3 retrievals for all RAG baselines using the format '*Context: {concatenated retrievals}*' in the beginning. The exact format for each baseline can be found in the Appendix A.

RAG. Embedding all Wikipedia documents in the database requires significant computation, so we randomly select 100K articles, including the articles used for dataset collection. We create separate databases for each genre. For soccer, we select articles linked to *Category:Association football*, while for baseball, we select from *Baseball*, *Basketball*, and *American football* to ensure enough articles. We parse each article using the LangChain document loader (see LangChain Documentation), and index the documents using FAISS (Johnson et al., 2019) following Shi et al. (2023). We chunk the documents with a maximum of 1500 characters, ensuring a 10-character overlap between chunks.

Metrics. To evaluate generated answers, we use the F1 score following Petroni et al. (2021). Since most answers in our QA dataset are either a noun phrase or a short sentence, we only consider the first sentence of each answer. We parse this first sentence using the nltk sentence tokenizer.[2]

5.2 Experimental Results

We present the performance of our entity linking and entity-centric QA tasks in the soccer genre. Additionally, the performance in the baseball genre is reported in Appendix B.

[2]https://www.nltk.org/api/nltk.tokenize.sent_tokenize.html

5.2.1 Results of Entity Linking

Table 3 presents the accuracy of our entity linking task in the soccer genre. To rigorously evaluate the performance of each method in resolving mentions across different levels of lexical similarity, we present the results for each bin of lexical similarity separately. For each mention, we calculate the Jaccard similarity (Zhang et al., 2021) with the entity name using a character-level token as a straightforward measure of lexical overlap. The number of mentions for each bin is presented in Table 4.

The results reveal a clear trend: as Jaccard similarity increases, the accuracy of all baselines improves. This underscores the importance of lexical similarity in entity linking tasks, where emerging mentions with lower lexical similarity typically lead to less accurate linking of mentions.

c-SpEL surpasses SpEL in 1112, but falls below in 0102 and 0304 in total. This implies that the performance of c-SpEL deteriorates as it moves further from the last time point of learning. Besides, TempCCA consistently outperforms other baselines in most cases, except for Set 4 (0.6 - 0.8), when ArboEL exceeds TempCCA from 0.11 to 0.4. Nevertheless, TempCCA shows a robust performance across all months and sets. Notably, TempCCA surpasses the baselines the most in Set 1, from 1.42 in 0102 to 8.27 in 1112. This implies that utilizing the recently predicted mentions can help jointly resolve mentions with low lexical overlap, as there tend to be similar mentions within a similar time step.

5.2.2 Results of Entity-Centric QA

Table 5 presents the performance of our entity-centric QA task in the soccer genre. The accuracy of TempCCA's top-1 entity linking prediction on QA questions attains 66.62, 67.62, and 65.86 for 1112, 0102, and 0304, respectively. Compared to the base LLM, LLM-ER improves performance by an average of 3, indicating that resolving new mentions helps the LLM generate more accurate responses. Still, both LLM and LLM-ER underperform the RAG baselines. Specifically, RaLM improves the LLM's performance by an average of 15 using the retrieved documents. Interestingly, RaLM-CoT performs significantly worse than RaLM by an average of 6.8, suggesting that entity prediction by the LLM itself does not effectively contribute to QA accuracy. On the other hand, RaLM-ER improves RaLM by an average of 1.4 and outperforms all other baselines, indicating that resolving mentions can be beneficial to knowledge-intensive QA tasks.

	1112	0102	0304
Average			
LLM	28.82	27.84	29.75
LLM-ER (Ours)	31.47	32.02	32.15
RaLM	44.48	42.75	46.55
RaLM-CoT	38.09	36.43	38.56
RaLM-ER (Ours)	**45.67**	**44.60**	**47.93**
Retrieval Hit			
LLM	28.48	28.00	31.33
LLM-ER (Ours)	32.91	33.61	34.74
RaLM	59.07	56.37	59.31
RaLM-CoT	50.55	48.01	48.03
RaLM-ER (Ours)	**59.42**	**56.71**	**60.24**
Retrieval Miss			
LLM	29.23	27.64	27.42
LLM-ER (Ours)	**29.69**	**30.09**	28.33
RaLM	26.60	26.30	27.74
RaLM-CoT	22.82	22.45	24.59
RaLM-ER (Ours)	28.84	29.98	**29.76**

Table 5: Results of entity-centric QA for each time segment in F1 scores. Our approach is applied to both LLM and RAG framework.

Retrieval	Hit		Miss	
Entity Linking	Success	Failure	Success	Failure
RaLM	58.78	**60.43**	29.86	23.42
RaLM-ER (Ours)	**60.01**	59.48	**32.12**	**25.29**

Table 6: Comparison of RaLM and RaLM-ER performance in retrieval hits and misses, and entity linking successes and failures.

To analyze the results comprehensively, we report the results separately for cases of retrieval success (retrieval hit) and failure (retrieval miss). As shown in the middle section of Table 5, RAG baselines significantly enhance LLM performance; RaLM improves the base LLM by 30.73, 28.37, and 27.98 in 1112, 0102, and 0304, respectively. Additionally, RaLM-ER performs on par with or slightly better than RaLM when the retrieval succeeds, despite potential errors in entity resolution. Conversely, when retrieval fails, RaLM performs worse than the base LLM, with a decrease of 3.16 in 1112. This highlights the critical impact of retrieval failures. On the other hand, LLM-ER enhances the base LLM in all cases. Furthermore, RaLM-ER mitigates hallucinations, improving RaLM by approximately 2.06, 3.58, and 2.02 in 1112, 0102, and 0304, respectively. Remarkably, RaLM-ER even outperforms all baselines despite incorrect

retrievals in 0304.

To further pinpoint the improvement, we analyze the results of RaLM and RaLM-ER in four scenarios: when entity resolution is correct or incorrect, within the context of both retrieval hits and misses. Table 6 shows the averaged results across time segments. When retrieval is successful, the performance of RaLM-ER improves by 1.2 when the entity resolution is correct; however, it drops approximately 1.0 when the entity resolution is wrong. Conversely, when retrieval fails, performance is enhanced in both correct and incorrect resolution cases. RaLM-ER improves upon RaLM by 2.3 when the resolution is correct and by 1.8 when the resolution is incorrect. Although it seems crucial to avoid introducing incorrect resolutions, providing entity resolution results to the LLM generally improves end-to-end performance.

6 Conclusion

In this work, we addressed the challenge of resolving new linguistic expressions in the dynamic and ever-evolving landscape of human language. We introduced DYNAMICER to evaluate the ability of models to resolve emerging mentions. Our benchmark proposes entity linking tasks for resolving emerging mentions and entity-centric QA tasks for RAG evaluation. To address the temporal dynamics of emerging mentions, we proposed a temporal segmented clustering method with continual adaptation. Our exhaustive experiments demonstrated that our method surpassed existing baselines in resolving new expressions, particularly when there is less lexical overlap.

Future work may extend to updating KB using entity resolution, which can directly handle the retrieval failures caused by mention dynamics. Through DYNAMICER, we provide a resource for the research community to further explore and improve dynamic entity resolution. We hope this fosters further research towards developing more robust models capable of handling the continuous emergence of new expressions.

Limitations

We acknowledge several limitations in our work. Firstly, our dataset and method are primarily designed to handle variations in single entity mentions and may not effectively address cases where multiple entities are combined into a single mention, such as "Kimye" referring to Kanye West and Kim Kardashian. Future research could explore developing benchmarks and models capable of resolving such combined mentions into multiple entities. Additionally, our dataset may reflect biases introduced by the GPT-4 model and the specific prompts used during its creation. Although we perform thorough human validation and revisions, future studies could benefit from employing a diverse set of language models to mitigate these potential biases.

Ethics Statement

Safety. All data were sourced from Tumblr, which is publicly available. To ensure the safety of our dataset, we conduct a two-stage filtering process. Initially, annotators were instructed to report any potentially harmful or privacy-invading content. Following this, the authors reviewed the remaining content to further filter out inappropriate materials. Despite these efforts, biases such as stereotyping may still be present due to the nature of real communication on Tumblr, where a significant portion of the user base consists of teenagers and young adults.

Intended Use. The DYNAMICER dataset is intended to be used for research purposes only, and the use is subject to Tumblr Terms of Service and Community Guidelines.

Annotator Compensation. We hired university students as annotators. To uphold ethical standards, we compensated our annotators with a fair hourly wage of approximately USD $15. The estimated completion time for each task was determined through multiple preliminary trials conducted by our research team. Consequently, the average expense per datapoint amounted to approximately $0.30. Data points requiring additional time were compensated at a proportionately higher rate to ensure fairness. The full text of instructions is provided in Figure 16.

Acknowledgements

We thank Wonkwang Lee, Chris Dongjoo Kim, Sangwoo Moon, Sehun Lee, Jongchan Noh, and the anonymous reviewers for their insightful discussions. This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2022-II220156, Fundamental research on continual meta-learning for quality enhancement of casual videos and their

3D metaverse transformation), the SNU-Global Excellence Research Center establishment project, the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2023R1A2C2005573), Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(RS-2023-00274280), and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)). Gunhee Kim is the corresponding author.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Dhruv Agarwal, Rico Angell, Nicholas Monath, and Andrew McCallum. 2022. Entity linking via explicit mention-mention coreference modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4644–4658, Seattle, United States. Association for Computational Linguistics.

Emily Allaway, Shuai Wang, and Miguel Ballesteros. 2021. Sequential cross-document coreference resolution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4659–4671.

Anietie Andy, Mark Dredze, Mugizi Rwebangira, and Chris Callison-Burch. 2017. Constructing an alias list for named entities during an event. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 40–44, Copenhagen, Denmark. Association for Computational Linguistics.

Rico Angell, Nicholas Monath, Sunil Mohan, Nishant Yadav, and Andrew McCallum. 2021. Clustering-based inference for biomedical entity linking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2598–2608, Online. Association for Computational Linguistics.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Nicholas Botzer, Yifan Ding, and Tim Weninger. 2021. Reddit entity linking dataset. *Information Processing & Management*, 58(3):102479.

Avi Caciularu, Arman Cohan, Iz Beltagy, Matthew E Peters, Arie Cattan, and Ido Dagan. 2021. Cdlm: Cross-document language modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2648–2662.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4545–4552.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Bhuwan Dhingra, Jeremy R Cole, Julian Martin Eisenschlos, Dan Gillick, Jacob Eisenstein, and William Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark. Association for Computational Linguistics.

Zhaochen Guo and Denilson Barbosa. 2018. Robust named entity disambiguation with random walks. *Semantic Web*, 9(4):459–479.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Julia Hirschberg and Christopher D Manning. 2015. Advances in natural language processing. *Science*, 349(6245):261–266.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 782–792.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606, Toronto, Canada. Association for Computational Linguistics.

Jungo Kasai, Keisuke Sakaguchi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, Kentaro Inui, et al. 2024. Realtime qa: What's the answer right now? *Advances in Neural Information Processing Systems*, 36.

Dayoon Ko, Jinyoung Kim, Hahyeon Choi, and Gunhee Kim. 2024. Growover: How can llms adapt to growing real-world knowledge? *Preprint*, arXiv:2406.05606.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Phong Le and Ivan Titov. 2018. Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1595–1604, Melbourne, Australia. Association for Computational Linguistics.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE transactions on knowledge and data engineering*, 34(1):50–70.

Ying Lin, Chin-Yew Lin, and Heng Ji. 2017. List-only entity linking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 536–541, Vancouver, Canada. Association for Computational Linguistics.

Adam Liska, Tomas Kocisky, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, D'Autume Cyprien De Masson, Tim Scholtes, Manzil Zaheer, Susannah Young, et al. 2022. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models. In *International Conference on Machine Learning*, pages 13604–13622. PMLR.

Lajanugen Logeswaran, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, Jacob Devlin, and Honglak Lee. 2019. Zero-shot entity linking by reading entity descriptions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3449–3460.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822.

Katerina Margatina, Shuai Wang, Yogarshi Vyas, Neha Anna John, Yassine Benajiba, and Miguel Ballesteros. 2023. Dynamic benchmarking of masked language models on temporal concept drift with multiple views. *arXiv preprint arXiv:2302.12297*.

Sunil Mohan and Donghui Li. 2018. Medmentions: A large biomedical corpus annotated with umls concepts. In *Automated Knowledge Base Construction (AKBC)*.

Sumit Neelam, Udit Sharma, Hima Karanam, Shajith Ikbal, Pavan Kapanipathi, Ibrahim Abdelaziz, Nandana Mihindukulasooriya, Young-Suk Lee, Santosh Srivastava, Cezar Pendus, Saswati Dana, Dinesh Garg, Achille Fokoue, G P Shrivatsa Bhargav, Dinesh Khandelwal, Srinivas Ravishankar, Sairam Gurajada, Maria Chang, Rosario Uceda-Sosa, Salim Roukos, Alexander Gray, Guilherme Lima, Ryan Riegel, Francois Luus, and L V Subramaniam. 2022. SYGMA: A system for generalizable and modular question answering over knowledge bases. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3866–3879, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2021. Kilt: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint conference on EMNLP and CoNLL-shared task*, pages 1–40.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *arXiv preprint arXiv:2302.00083*.

Christopher Sciavolino, Zexuan Zhong, Jinhyuk Lee, and Danqi Chen. 2021. Simple entity-centric questions challenge dense retrievers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6138–6148.

Hassan Shavarani and Anoop Sarkar. 2023. SpEL: Structured prediction for entity linking. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11123–11137, Singapore. Association for Computational Linguistics.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the gap: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.

Nishant Yadav, Ari Kobren, Nicholas Monath, and Andrew Mccallum. 2019. Supervised hierarchical clustering with exponential linkage. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6973–6983. PMLR.

Klim Zaporojets, Lucie-Aimée Kaffee, Johannes Deleu, Thomas Demeester, Chris Develder, and Isabelle Augenstein. 2022. Tempel: Linking dynamically evolving and newly emerging entities. *Advances in Neural Information Processing Systems*, 35:1850–1866.

Peiying Zhang, Xingzhe Huang, Yaqi Wang, Chunxiao Jiang, Shuqing He, and Haifeng Wang. 2021. Semantic similarity computing model based on multi model fine-grained nonlinear fusion. *IEEE Access*, 9:8433–8443.

A Experimental Details

Dataset generation. When we use GPT to generate dataset, we use gpt-4-turbo and set the max_token as 1024 and the temperature 0.

EL Baseline Training. For SpEL, we use `roberta-base` for the encoder and conduct only the final fine-tuning step based on the second-step pretrained model. We train the model for 10 epochs while setting the batch_size to 16, bert_dropout to 0.2, and label_size to 10240. We report the results of the micro entity linking metrics. For ArboEL, we set the batch_size to 32 and use the other hyperparameters the same as in the ArboEL setting without any modifications.

TempCCA. For each training time step, we train the bi-encoder (`bert-base-cased`) (Devlin et al., 2018) model for 5 epochs with a batch size of 32 and a learning rate of 3e-05 using the Adam optimizer. In each training iteration, we randomly select 30 mentions to form an entity cluster if the number of previously resolved mentions for that entity exceeds 30. We set the hyperparameter α as 0.8. It requires A6000 X 4 GPUs for training and takes 6 hours for continual training.

RAG. We use e5-base for the retriever from hugging face `intfloat/e5-base`. We set *temperature* and *max_new_tokens* to 0.3 and 30, respectively, for all baselines except the first text generation in RaLM-CoT, for which we set *temperature* to 0.1 and *new_tokens* to 10 to ensure accurate entity prediction.

QA. The prompt formats for five different QA baselines are presented below.

```
# LLM
Given a question, please provide a short answer.
Question: {question}
Answer:

# LLM-ER
The mention {mention} may also be referred to as {entity}.
Given a question, please provide a short answer.
Question: {question}
Answer:

# RaLM
Context: {context}
Given a question, please provide a short answer.
Question: {question}
Answer:

# RaLM-CoT
## first
Context: {context}
Question: {question} {mention} is
## second
Context: {context}
Question: {quesition} {mention} is {first answer}.
Answer:

# RaLM-ER
Context: {context}
The mention {mention} may also be referred to as {entity}.
Given a question, please provide a short answer.
Question: {question}
Answer:
```

Table 7: Prompt template for QA task

B Additional Experimental Results

Results for the Baseball Genre. We report the performance of our entity linking and entity-centric QA task in the baseball genre. As shown in Table 8, TempCCA demonstrates robust performance across all months. TempCCA exceeds ArboEL from 2.14 in 0102 to 3.54 in 0304. In Table 9, LLM-ER exceeds the base LLM by an average of 1.6. RaLM consistently enhances the LLM's performance by an average of 16 using the retrieved documents. RaLM-ER improves RaLM by an average of 0.8 and outperforms all other baselines. In cases of retrieval success (retrieval hit), the RAG baselines significantly boost LLM performance. In this case, RaLM-ER demonstrates the best performance. On the contrary, when retrieval fails, RaLM performs worse than the base LLM. However, LLM-ER still improves upon the base LLM in these cases. RaLM-ER effectively mitigates hallucinations, improving RaLM by 1.14, 0.85, and 1.12 in 1112, 0102, and 0304, respectively.

Method	1112	0102	0304
SpEL	59.93	53.99	67.80
c-SpEL	57.45	53.07	66.59
ArboEL	85.38	85.93	85.66
TempCCA	**88.26**	**88.07**	**89.20**

Table 8: Results of the entity linking task by lexical similarity and time segment.

	1112	0102	0304
Average			
LLM	30.25	31.92	28.38
LLM-ER (Ours)	31.59	34.54	29.16
RaLM	44.54	50.45	45.06
RaLM-CoT	36.92	42.76	38.77
RaLM-ER (Ours)	**45.56**	**51.13**	**45.70**
Retrieval Hit			
LLM	30.05	30.65	30.38
LLM-ER (Ours)	32.64	33.84	31.37
RaLM	63.27	63.18	62.34
RaLM-CoT	50.94	53.14	51.89
RaLM-ER (Ours)	**64.15**	**63.73**	**62.58**
Retrieval Miss			
LLM	30.44	33.62	25.94
LLM-ER (Ours)	**30.63**	**35.48**	**26.46**
RaLM	27.30	33.44	23.95
RaLM-CoT	24.01	28.90	22.74
RaLM-ER (Ours)	28.44	34.29	25.07

Table 9: Results of entity-centric QA for each time segment in F1 scores. Our approach is applied to both LLM and RAG framework.

Additional Factual Accuracy Evaluation. To ensure robustness on evaluating entity-centric QA tasks, we conduct an additional evaluation using GPT-4 to assess factual accuracy. Following the methodology of Kamalloo et al. (2023), we prompt GPT-4 to assess factual accuracy and detect potential hallucinations in candidate answers using the following zero-shot prompt: '*Question: {question}, Answer: {gold answer}, Candidate: {prediction}: Is candidate correct?*'. Table 10 presents the results in the soccer genre. Consistent with the results of F1 scores, our LLM-ER method improves upon the standard LLM, and our RaLM-ER significantly enhances RaLM performance.

GPT-4 score	1112	0102	0304
LLM	26.37	23.86	30.47
LLM-ER (Ours)	26.53	23.70	30.93
RaLM	59.70	62.50	63.43
RaLM-CoT	57.14	60.95	57.05
RaLM-ER (Ours)	**62.85**	**63.80**	**66.59**

Table 10: Factual accuracy results for entity-centric QA tasks assessed by GPT-4.

Analysis of Entity-Centric QA task using Jaccard similarity. We report that lower Jaccard similarity between mentions and entity names degrades entity linking performance. To further investigate this, we analyze the entity-centric QA task using Jaccard similarity to observe how the lexical overlap of mentions in questions affects the answer generation performance. Table 11 shows the results in the soccer genre. Our finding indicates that as Jaccard similarity increases, the end-to-end generation accuracy also improves in the QA task. This shows that the RAG performance may degrade with variational mention surface forms. Notably, the score gap between RaLM and RaLM-ER increases as the Jaccard similarity decreases. It implies that effectively resolving emerging mentions is critical to enhance the end-to-end performance.

	Set 1 (0.0-0.2)	Set 2 (0.2-0.4)	Set 3 (0.4-0.6)	Set 4 (0.6-0.8)	Set 5 (0.8-1.0)
RaLM	29.90	32.91	43.32	46.04	47.75
RaLM-ER	34.98	35.46	44.80	48.20	48.36

Table 11: Analysis of entity-centric QA task using Jaccard similarity

Candidate Generation Results. We also report the Recall@n score in the soccer genre for ArboEL and TempCCA in Table 12. Recall@n score considers a prediction successful if the correct entity appears within the top-n predicted entities. TempCCA consistently outperforms ArboEL regardless of top-n, which indicates that our setting of temporal segmented clustering surpasses the baseline performance in entity candidate generation as well.

Recall@	1	2	4	8	16	32	64
1112							
ArboEL	87.67	90.01	91.74	92.95	94.26	95.13	95.90
TempCCA (Ours)	**88.96**	**90.80**	**92.51**	**93.62**	**94.60**	**95.46**	**96.19**
0102							
ArboEL	84.67	87.69	89.78	91.44	92.82	93.98	**95.18**
TempCCA (Ours)	**86.19**	**88.70**	**90.71**	**92.16**	**93.31**	**94.18**	**95.18**
0304							
ArboEL	86.03	88.91	90.87	92.19	93.38	94.56	95.58
TempCCA (Ours)	**87.00**	**89.54**	**91.34**	**92.77**	**93.93**	**94.91**	**95.60**

Table 12: Results of candidate generation

C Tag List

Below are the tags used to scrape the documents:

Soccer genre: #AC Milan, #Arsenal FC, #Atletico Madrid, #Borussia Dortmund, #Bundesliga, #Chelsea FC, #FC Barcelona, #FC Bayern, #Juventus, #La Liga, #Ligue 1, #Liverpool FC, #Manchester City, #Manchester United, #Premier League, #PSG, #Real Madrid, #Serie A, #Tottenham Hotspur, #UCL, #West Ham

Baseball genre: #Arizona Diamondbacks, #Atlanta Braves, #Baltimore Orioles, #Boston Red Sox, #Chicago Cubs, #Chicago White Sox, #Cincinnati Reds, #Cleveland Guardians, #Colorado Rockies, #Detroit Tigers, #Houston Astros, #Kansas City Royals, #Los Angeles Angels, #Los Angeles Dodgers, #Miami Marlins, #Milwaukee Brewers, #Minnesota Twins, #New York Mets, #New York Yankees, #Oakland Athletics, #Philadelphia Phillies, #Pittsburgh Pirates, #San Diego Padres, #San Francisco Giants, #Seattle Mariners, #St. Louis Cardinals, #Tampa Bay Rays, #Texas Rangers, #Toronto Blue Jays, #Washington Nationals

D QA Generation Details

In this section, we describe the details of QA generation. We first filter out ambiguous mentions using GPT-4 as described in Section 3.4. For the remaining mentions, we present a few examples as shown in Table 14 and provide each mention to

three annotators. They are tasked with distinguishing between ambiguous and unambiguous mentions. Only mentions judged as unambiguous by at least two annotators are retained, and the rest are filtered out.

Following the procedure of initial generation from Ko et al. (2024), we then generate entity-centric QA pairs. First, we gather articles related to the entity referred to by each mention. For each article, we extract a paragraph between 300 and 2000 characters in length and perform random sampling based on the number of unambiguous mentions referring to the entity. We then provide GPT-4 with the prompt shown in Table 15 to generate the QA pairs. To simultaneously evaluate the success of retrieval model in the RAG framework, we annotate the evidence text required to answer the generated questions correctly. The evidence text is the paragraph given as input to GPT-4. By following this method, we create entity-centric QA pairs, and subsequently, we replace the entity names in the questions with the corresponding mentions. After generating QA pairs, authors validate whether the generated QA pairs are answerable given questions, filtering out the pairs if not.

E Training Procedure

We adopt Agarwal et al. (2022)'s training procedure, which shows state-of-the-art performance on the MedMentions dataset (Mohan and Li, 2018). In this section, we summarize Agarwal et al. (2022)'s procedure, including positive and negative sampling. The goal is to optimize the affinity functions by using batch-wise gradient optimization. For each incoming batch of mentions M_B, we construct a graph G_{M_B}, where the nodes represent each mention $m_i \in M_B$, mentions coreferent with m_i, and the that include each mention $m_i \in M_B$, mentions coreferent with m_i, and the corresponding set of gold entities for each mention in the batch. Consequently, the full set of edges in the graph G_{M_B} for batch is represented as:

$$E(G_{M_B}) = \bigcup_{m_i \in M_B} \Big(\{(e_i^*, m_k) \mid m_k \in \mathcal{S}_{e_i^*}\} \cup \{(m_k, m_l) \mid m_k, m_l \in \mathcal{S}_{e_i^*}\} \Big) \tag{3}$$

Here, $\mathcal{S}_{e_i^*}$ represents the set of mentions coreferent with the gold entity e_i^*.

Positive and Negative Sampling

The graph G_{M_B} is partitioned into disjoint clusters. The constraints for each cluster C are:

1. C includes at most one entity.
2. For all $u, v \in C$, if u is connected to v, then $f(u, v) \leq \lambda$,
3. For all $u, v \in C$, either u is connected to v or v is connected to u.

Following Angell et al. (2021); Agarwal et al. (2022), we sort edges by decreasing dissimilarity, and iteratively remove edges. First, we eliminate all edges in graph G with weights exceeding λ. Next, we process each edge $(u, v) \in E$ in descending order of dissimilarity, checking if its presence violates any of the three defined constraints. If an edge does violate a constraint, we remove it from E. If not, we examine whether the connected component of node u contains an entity, i.e., $|C_u \cap \mathcal{E}| = 1$. If it does, we drop edge (u, v) if v can still be reached by an entity node without (u, v). We retain edge (u, v) and continue iteration if not reachable, preserving the connectivity of the cluster. Our predicted clusters are the final connected components in graph G.

For negative sampling, we identify negative edges for each mention $m_i \in B$. The negative edges include the $k/2$ lowest-weight incoming edges from each of $\mathcal{E} \setminus \{e_i^*\}$ and $\mathcal{M} \setminus \mathcal{S}_{e_i^*}$.

Loss Function

Following Yadav et al. (2019); Agarwal et al. (2022), the loss function $\mathcal{L}(m_i)$ is defined as follows:

$$\mathcal{L}(m_i) = \sum_{p \in \kappa(m_i)} (I_{p,m_i} \log(\sigma(w_{p,m_i})) + (1 - I_{p,m_i}) \log(1 - \sigma(w_{p,m_i}))) \tag{4}$$

where $\kappa(m_i)$ includes all neighbors with outgoing edges to m_i in the graph, I_{p,m_i} is an indicator variable, with $I_{p,m_i} = 1$ if (p, m_i) is in a pruned set of edges; otherwise, $I_{p,m_i} = 0$. $\sigma(\cdot)$ denotes the softmax function. The total loss for the batch B is the average of the losses for all mentions in B, optimizing for higher probabilities for positive edges and lower probabilities for negative edges.

F Dataset Generation Prompts

Table 13 shows the prompt used for the mention detection from the corpora. Table 14 shows the prompt used for filtering ambiguous mentions, which is not suitable for generating entity-centric QA. Table 15 shows the prompt used for generating entity-centric QA.

G Case Study

Tables 17, 18, and 19 present a case study from entity-centric QA.

Please identify all expressions in the given text that explicitly name or describe a player, coach, or team. This includes direct names, nicknames, and any role-specific references (like positions or accolades) that imply a particular individual or team.

Instructions:

1) Provide a list of expressions (ex. [“our striker”, “messi”, “agent of chaos”, ...])

2) The expressions can be long, as they also include noun phrases.

3) Write expressions exactly as they appear in the text. Do not modify the expressions in the text, even if they contain typos. Maintain the original capitalization and spacing, as we will use string matching.

4) List expressions in the order they appear in the text. If an expression appears multiple times, list it multiple times in the order of appearance.

5) If the text lacks a suitable expression, provide an empty list.

Be sure to follow the following format and write your answer within the list: [“Expression 1”, “Expression 2”, ...]

Text: {*text*}

Table 13: Sample prompt for mention detection

Provided List: {*mention list*}

The provided list is a compilation of mentions identified in the textual corpora. Upon human verification, these mentions refer to {*entity*} in the text.

Select the mentions from the list that unambiguously refer to without context.

Unambiguous mentions are mentions that exclusively refer to the specified entity ({*entity*}) without requiring additional context.

Examples:

Unambiguous Mentions for MANCHESTER_UNITED_F.C.: “Man Utd”, “20-time English champions”, “the Red Devils club”, ...

Ambiguous Mentions for MANCHESTER_UNITED_F.C.: “United”, “chaos club”, “English Powerhouse”, “the First Team”, ...

Instructions:

1) Write mentions exactly as they appear in the provided list. Do not modify the mentions in the list, even if they contain typos. Maintain the original capitalization and spacing.

2) Be sure to follow the following format and write your answer within the list: [“Mention 1”, “Mention 2”, ...]

Table 14: Sample prompt for filtering ambiguous mentions

Generate a Q&A pair about [{*entity*}] based on a given context. The context will provide factual information about [{*entity*}].

Assume the person answering the question has common sense and is aware of the details and key points in the context, but the context itself is not quoted or referenced directly.

Context: {*context*}

Follow these instructions to generate a Q&A pair:

1) Provide a question and an answer.

2) Bracket the corresponding Entity ([{*entity*}]) in the Question, like the sample Q&A pair given below.

3) Do NOT use phrases such as ‘according to the context’ in your question.

4) Generate a SINGLE Q&A pair.

5) Provide a SHORT ANSWER.

Write your Q&A pair within curly brackets using the following format: {Question}{Answer}

Sample Q&A pair : {Where was [Lionel_Messi] born?}{Lionel Messi was born in Rosario, Argentina.}

Table 15: Sample prompt for generating entity-centric QA pairs

This context involves a social media post regarding sports: Association football (Baseball).
Please review the expressions that GPT-4 highlighted, which include direct names, nicknames, and role-specific references implying a specific individual or team.
If any of the highlighted expressions are incorrect, please revise them.
After revising, link each expression to its corresponding Wikipedia entity. You can search for these entities on Wikipedia and provide the URL. If the context is too ambiguous to resolve, or no suiable Wikipedia entity exists, label it as NOT VALID.
The provided mention *{mention}* refers to *{entity}* in the text.
Please determine whether the mention unambiguously refer to *{entity}* without context.

Table 16: Full text of instructions

Question: When did *Targett* make his first Premier League start?
Gold Answer: December 28, 2014.
Gold Entity: Matt Targett
LLM: **1992**. (F1 score: 0)
LLM-ER: Targett made his first Premier League start for Southampton against Manchester United on **August 16, 2014**. (F1 score: 0.11 / Entity Linking: Succeed)
RaLM: Targett made his first Premier League start for Aston Villa on **23 September 2014**. (F1 score: 0.12 / Retrieval: Hit)
RaLM-CoT: Targett made his first Premier League start for Aston Villa on **23 September 2014**, in a 2-1 victory at ... (F1 score: 0.09 / Retrieval: Hit)
RaLM-ER: Targett made his first Premier League start for Southampton on **28 December 2014**, in a 1–1 draw against Chelsea. (F1 score: 0.29 / Retrieval: Hit / Entity Linking: Succeed)

Table 17: Case: Properly retrieved document with correctly resolved entity synergies in RaLM-ER.

Question: When did *Caoimhin the Prettiest of all Kelleher* sign his long-term deal with the club?
Gold Answer: Caoimhín Kelleher signed his long-term deal on 24 June 2021.
Gold Entity: Caoimhín Kelleher
LLM: Caoimhin Kelleher has not signed a long-term deal with the club. (F1 score: 0.42)
LLM-ER: Caoimhin the Prettiest of all Kelleher signed his long-term deal with the club in 2021. (F1 score: 0.52 / Entity Linking: Succeed)
RaLM: In August 2018. (F1 score: 0 / Retrieval: Miss)
RaLM-CoT: In August 2018, Caoimhín Odhrán Kelleher signed a new contract with Liverpool. (F1 score: 0.29 / Retrieval: Miss)
RaLM-ER: Caoimhin the Prettiest of all Kelleher signed his long-term deal with the club in 2021. (F1 score: 0.52 / Retrieval: Miss / Entity Linking: Succeed)

Table 18: Case: Correctly resolved entity refrains from hallucinations in RaLM-ER.

Question: What record did *Trentnation* set after Liverpool's victory in the 2022 FA Cup Final?
Gold Answer: He became the youngest-ever player to win six major trophies for an English club at the age of 23 years and 219 days.
Gold Entity: Trent Alexander-Arnold
LLM: Trent Alexander-Arnold set a record for being the youngest player to win the FA Cup, FA Cup Final Man of the Match, and FA Cup (F1 score: 0.29)
LLM-ER: Trentnation set the record for the most assists in a single season in the Premier League with 13 assists. (F1 score: 0.06 / Entity Linking: Succeed)
RaLM: None, as Trent Alexander-Arnold did not set any record in the 2022 FA Cup Final. (F1 score: 0 / Retrieval: Miss)
RaLM-CoT: There is no record set by Trentnation (Trent Alexander-Arnold) in the 2022 FA Cup Final. (F1 score: 0 / Retrieval: Miss)
RaLM-ER: Trent Alexander-Arnold became the youngest player to lift the FA Cup, at the age of 24. (F1 score: 0.36 / Retrieval: Miss / Entity Linking: Succeed)

Table 19: Case: Correctly resolved entity refrains from hallucinations in RaLM-ER.

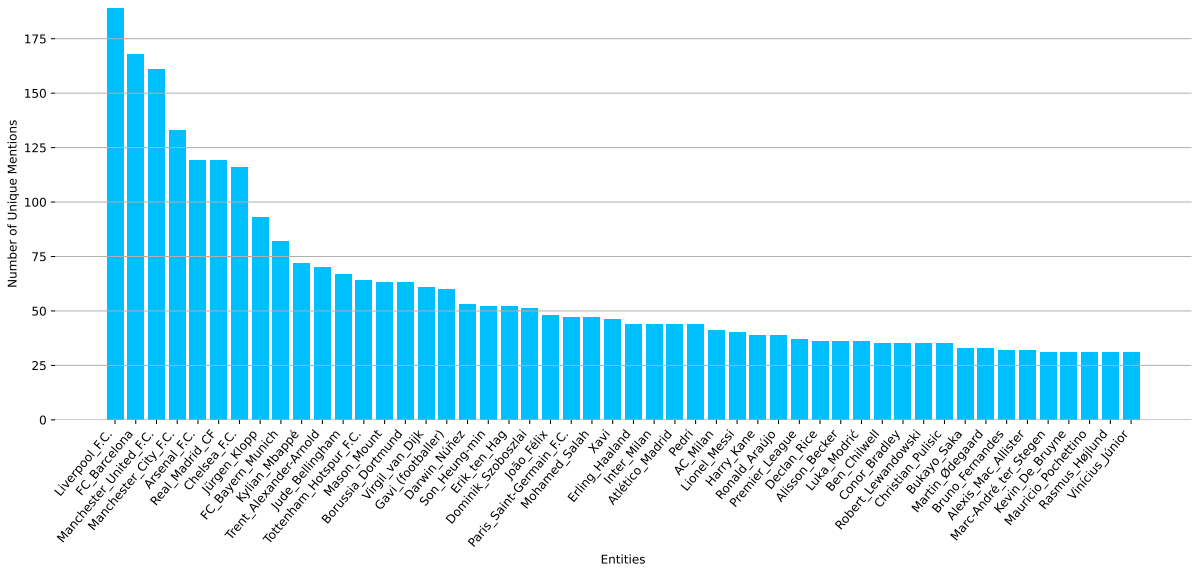

Figure 3: Mention variations in DYNAMICER