imEL: Instance-level Masked Entity Linking Model

Jaeeun Jang*, Sangmin Kim*‡, Mikyoung Lee*, Mira Yun†, and Charles Wiseman†
*AI Research Team, Hanwha Systems, Seongnam-Si, Gyeonggi-Do, South Korea
†Department of Computer Science, Boston College, Chestnut Hill, MA, USA
‡Corresponding author
Email: {wkdwodms0779,smkim0153,mk1013.lee}@hanwha.com, {yunmd,wisemacc}@bc.edu

2024 58th Annual Conference on Information Sciences and Systems (CISS) | 979-8-3503-6929-8/24/$31.00 ©2024 IEEE | DOI: 10.1109/CISS59072.2024.10480212

***Abstract*—Knowledge base question answering (KBQA) is a complicated natural language processing task. KBQA systems are used to answer questions about a large data set where the goal is to identify the entities in the question and link them to other relevant information in the same knowledge base. Prior entity linking (EL) systems inherently use a class-based approach whereby two similar entities distinguishable only through context are treated as a single entity. For example, two vehicles both mentioned as "blue SUVs" but with different time and location contexts will be treated as the same entity. This work introduces a new type of EL called instance-level EL that can recognize and utilize instance-specific information. One such instance-level EL model, the instance-level masked EL model (*imEL*) is described and evaluated. *imEL* has a high level of accuracy in responding to questions that require identification and linking of specific instances in the input text.**

***Index Terms*—Entity Linking, Deep Learning, Natural Language Processing.**

I. Introduction

The global market for intelligent virtual assistants is experiencing rapid growth. Many organizations and institutions are adopting chatbots as a cost-effective alternative to human customer service. Furthermore, a wide range of applications, including healthcare, educational platforms, finance, and technical support, have integrated knowledge base question answering (KBQA) systems to respond to user questions and provide information. These KBQA systems retrieve answers from a knowledge base (KB) containing entities and their relationships [1], [2]. Given the substantial data reservoir within the KB, it is crucial to accurately identify the appropriate entity in question and link it to other relevant entities stored within the KB. This process is known as entity linking (EL) in the field of natural language processing.

EL is the process of aligning mentions of entities in unstructured documents with their corresponding entities in a KB. This task presents challenges due to the frequent ambiguity of entity mentions within the document text. For instance, the term "Jordan" might refer to the country, the shoe brand, or the basketball player. Aligning the term to the appropriate KB entity, therefore, cannot rely solely on entity mentions. In many cases, a comprehensive understanding of the context surrounding these entity mentions is essential for achieving successful disambiguation and entity linking.

Current EL systems demonstrate performance surpassing that of humans on standard KBs such as Wikipedia. Nonetheless, they still encounter limitations when faced with more sophisticated linking requirements [3]–[9]. For instance, in Fig. 1, current EL systems would link both occurrences of *Boeing 747* in the given sentences to the entity "Boeing 747". This linkage occurs because these two instances of *Boeing 747* share the same type, even though they represent distinct aircraft with different departure and arrival regions. We categorize these EL approaches as class-level EL. In contrast, our proposed instance-level EL distinguishes between two entity mentions by incorporating *instance-specific information*, such as time, altitude, speed, location, and direction, which varies between instances. In this example, our instance-level EL system would link *Boeing 747* in Sentence1 to the entity "Boeing 747_[unique_id1]" and *Boeing 747* in Sentence2 to the entity "Boeing 747_[unique_id2]". A KBQA system may extract knowledge that is not relevant when it cannot precisely identify each entity in the question. This could lead to the delivery of low confidence answers to users. In this paper, we introduce an instance-level EL model designed for downstream tasks that require the identification of individual instances of similar entities.

Sentence1: At 14:51 on January 14, 2022, a Boeing 747 is flying over Incheon.
Sentence2: At 15:07 on January 14, 2022, a Boeing 747 is flying over Incheon.

Sentence	Entity Mention	Place of Departure	Place of Arrival
Sentence1	Boeing 747	Seoul	London
Sentence2	Boeing 747	California	Seoul

Fig. 1. Example input sentences given to an EL system.

We compile a KB enriched with battlefield reports crafted with Korean military experts. Our military QA system excels in addressing inquiries like *"Where did the fighter aircraft that was over the Yellow Sea around 2pm yesterday move to around 9am today?"*. Precise identification of the entity *"fighter aircraft"* in the question is crucial for the accurate functioning of our QA system. Consequently, we introduce *imEL: instance-level masked EL model*, designed to achieve advanced entity identification through in-depth interaction with instance-specific information.

The structure of this paper is as follows: Section II provides an overview of existing class-level EL methodologies and outlines their limitations. In Section III, we present the design and model architecture of our *imEL* system. The effectiveness of our solution is illustrated in Section IV, which includes

979-8-3503-6929-8/24/$31.00 ©2024 IEEE

details of the experimental setup and implementation. The paper concludes with future directions in the final section.

II. Related Work

While EL has been extensively researched to date, a common design choice in most current methods involves linking entity mentions to information in the KB, such as entity type, entity description, KB facts, etc., rather than directly connecting them to the corresponding entity names in the KB [3]–[9].

Several studies have employed entity types in the context of EL [5], [6]. For instance, *Bill Clinton*'s entity type is identified as "politician", and *England*'s entity type is labeled as "country". These studies address the EL challenge by framing it as a task of predicting entity types. The rationale behind this approach is that predicting entity types can lead to better performance, especially for rare entities, compared to directly predicting entity names.

Entity descriptions have been incorporated in recent EL studies [3], [4]. In these approaches, the first few sentences of Wikipedia entries serve as entity descriptions. For example, the Wikipedia entity *Michael Jordan* would be accompanied by the entity description: "Michael Jeffrey Jordan is an American former professional basketball player and businessman." These studies highlight the importance of encoding not just entity mentions but also entity descriptions, emphasizing the significance of achieving superior performance, even on test sets with distributions differing from the training set. Furthermore, Ayoola et al. [8] employed both entity types and entity descriptions to perform linking.

In order to perform linking even for entities with insufficient or missing entity types and descriptions, Ayoola et al. [9] employs KB facts to train the model. For instance, considering *Bill Clinton*'s birthplace is *Hope, Arkansas*, a relationship labeled *[place of birth]* is established between *Bill Clinton* and *Hope, Arkansas*, resulting in the KB fact *[Bill Clinton, place of birth, Hope/Arkansas]*. By incorporating a more diverse set of information, their model demonstrated superior performance compared to existing models that rely on a single source of information.

In contrast to the methods mentioned above, Cao et al. [7] refrains from utilizing KB information. Furthermore, rather than adopting the classifier approach employed by the aforementioned studies, it employs a sequence-to-sequence structure, treating the EL task as a translation problem. This new approach enables the model to capture fine-grained interactions between the input and entity names [7], [10]. As demonstrated in [9], the sequence-to-sequence model excels beyond methods reliant on KB information.

There are two limitations when applying the previously mentioned class-level EL methods to our military QA system. Firstly, the majority of EL research utilizes Wikipedia as the KB because it provides diverse information related to entities, including descriptions, types, and facts. Most class-level EL methods enhance their performance by leveraging information stored in the KB due to the abundant sources available for disambiguating entities. Many applications and systems, however, lack such context-rich KB information. For example, our military KB is derived from battlefield reports where it is not always feasible to record and manage contextual information around each individual instance. Secondly, all class-level EL methods overlook instance-specific information. This information is crucial in the context of instance-level EL for the model to be effectively utilized. In this paper, we propose the first instance-level EL method that can be universally employed in environments without accessible KB information. Additionally, we introduce a technique enabling the EL model to effectively learn instance-specific information.

III. *imEL: Instance-level Masked EL Model*

imEL stands as the pioneering instance-level EL model, crafted to achieve advanced entity identification through profound interaction with instance-specific information. Unlike existing class-level EL methods, *imEL* does not rely on KB information. *imEL* instead employs the superior sequence-to-sequence architecture [9] as the baseline model for the instance-level EL task proposed in this paper. The application of this sequence-to-sequence architecture to an instance-level EL task requires careful consideration to effectively capture the instance-specific information embedded in the sentences. Section III-B introduces our masking technique, *Masking Entity Mentions (MEM)*, designed to enable the model to make predictions after comprehensively understanding instance-specific information.

A. Resolving instance-level EL with a translation approach

Using a pre-trained sequence-to-sequence model for the EL task offers several advantages. First, it enables the fine-tuning of downstream-task data by utilizing the pre-trained language model (PLM) backbone without adding new layers that require learning from scratch [11]–[13]. Second, once the input and output formats of the model are determined, this approach can be universally applied to various instance-level EL tasks [14]. Third, the translation capability of the sequence-to-sequence model proves advantageous in instance-level EL. As the model's decoder generates the entity name autoregressively in subword token units, it interacts with the input context of the encoder at each time step [7], [10], [15]. For example, consider the sentence "At 07:49 on Apr 05, 2019, a MIG is flying at 6000ft over the Yellow Sea(33.03N, 125.90E)." The sequence-to-sequence model links the entity mention "MIG" to the target entity name "MIG21_[unique_id1]" through the following translation process. Initially, through the words "flying", "ft", and "above", the model roughly understands that the instance refers to a "aircraft". Subsequently, with the inclusion of the word "MIG", the model refines its understanding, specifying that the "aircraft" is produced by "Russian Corporation MIG". In contrast, classifier approaches might overlook these nuanced interactions as their final decision relies solely on the dot-product [7].

The EL task can be divided into two phases: mention detection (MD) and entity disambiguation (ED). MD involves

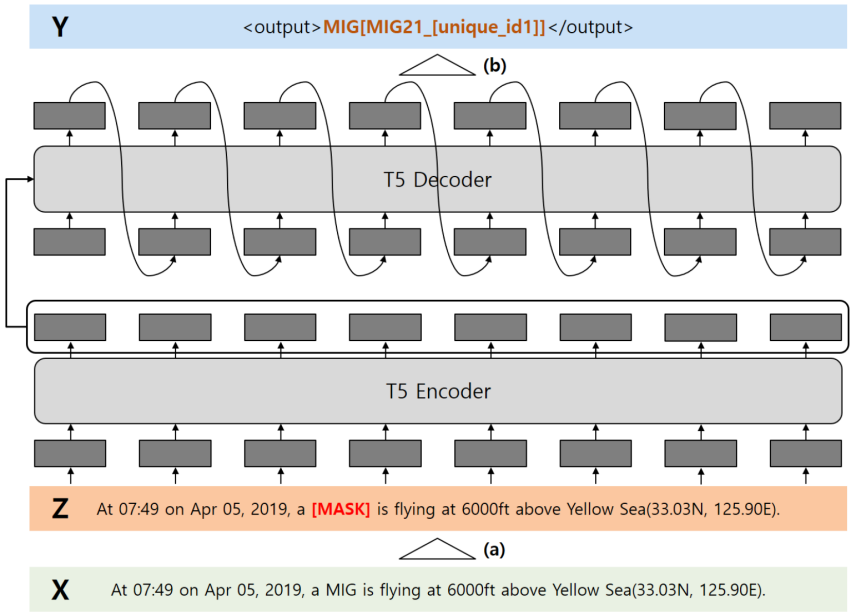

Fig. 2. *imEL* model architecture: (a) Applying "*MEM*" to the input X, (b) Generating entity mention and entity name in an autoregressive fashion.

recognizing entity mentions in the text, while ED aims to link each mention to a corresponding KB entity. As shown in Fig. 2, we designed our baseline model to generate the results of MD and ED in an autoregressive manner. Specifically, the baseline model utilizes a transformer-based sequence-to-sequence architecture that is pre-trained on a large corpus with a language modeling objective. It is then fine-tuned to generate the output Y conditioned on the input X. As shown in (1), the input X of our *imEL* consists of text only, without access to KB information, and the output Y includes both the entity mention and corresponding entity name.

$$\begin{aligned} X &= context_{left}\ ;\ Mention\ ;\ context_{right} \\ Y &=< output > Mention[Entity] < /output > \end{aligned} \tag{1}$$

where input X can have multiple entity mentions, all of which are encoded in a single forward pass.

B. Masking Entity Mentions (MEM)

In the example presented in Sec III-A and Fig. 2, our *imEL* model can infer that the instance is a "aircraft produced by Russian Corporation MIG" through interaction with information such as "flying", "ft", "above", and "MIG". It is possible to approach the correct answer even with a shallow interaction with the context. However, additional interaction with the input context is required for the model to output that the "aircraft produced by Russian Corporation MIG" is "MIG21", not "MIG23", "MIG25", or "MIG29". The model must be able to recognize detailed instance-specific information such as time, altitude, speed, location, and direction to make inferences at the instance level. Information that varies depending on the instance may be irrelevant in class-level EL but is crucial in instance-level EL. Fig. 1 illustrates that phrases such as "At 14:51 on January 14, 2022" and "Incheon" may not be important in inferring the target entity name for class-level EL. These words are key to revealing the identity of the instance mentioned in the sentence for instance-level EL. Nevertheless, the target entity name can often be inferred from entity mentions alone. The training process may end without the model learning deep interactions with such instance-specific information. Motivated by this, we propose *Masking Entity Mentions (MEM)* that allows models to interact deeply with instance-specific information by preventing excessive reliance on entity mentions.

The model must be trained to generalize well to unseen data in order to effectively utilize all available information. Enhancing the generalization performance of the model requires avoiding overfitting to specific features during training. Various techniques have been proposed to address overfitting. Dropout [16] encourages the network to learn from various combinations of neurons, reducing its reliance on specific neurons. This, in turn, helps prevent overfitting and enhances the model's generalization ability. Denoising auto-encoder [17] is trained to recover from a corrupted version of an input to a clean version. This is motivated by the goal of learning representations of the input that are robust to small irrelevant changes in input.

We propose a technique for randomly masking entity mentions in input sentences that is inspired by these two prior approaches. This prevents entity mentions from being used when generating entity names in the decoder, thereby allowing deeper interaction with the context and instance-specific information. *MEM* is depicted in Fig. 2. It shares similarities with Bidirectional Encoder Representations from Transformers (BERT)'s Masked Language Model (MLM) [13], albeit with a few distinctions. Firstly, the goals of masking in the two

methods differ. BERT's MLM aims to obtain contextualized word representations. *MEM* aims to prevent the model from excessively relying on specific information. Secondly, the input and output formats of BERT and *MEM* vary. BERT's masking target includes all input tokens such as "At", "Apr", "MIG", "flying", and "Sea". In *MEM*, only entity mentions such as "MIG" are subject to masking as shown in (2). Additionally, BERT is trained to recover only the original tokens from the corrupted version. *MEM*, in contrast, is trained to predict not only entity mentions but also entity names corresponding to the masked parts as shown in Y of (1).

$$M = context_{left}\ ;\ [MASK]\ ;\ context_{right} \tag{2}$$

where M is a form in which entity mentions in X are masked.

Masking should not be performed for all entity mentions during training as the $[MASK]$ token never appears in the inference time. We apply a "masking threshold" so that the model can sometimes get hints from entity mentions. As shown in (3), if p is smaller than the masking threshold p_m, mention masking is not applied to X. Otherwise, mention masking is applied to X. In Section IV-C, we present our test set performance with varying masking threshold values.

$$Z = \begin{cases} X & \text{if} \quad p < p_m \\ M & \text{if} \quad p \geq p_m \end{cases} \tag{3}$$

where p is sampled from a uniform distribution.

C. Optimization and Inference

There is a distinction in the application of *MEM* between optimization and inference. This follows as the model needs to utilize all information at inference time.

1) Optimization: The model is trained to maximize $\log p_\theta\left(Y|Z\right)$ with respect to model's parameters θ. This model is a sequence-to-sequence architecture, as shown in (4), so Y is sequentially generated token by token in an autoregressive manner conditioned on Z.

$$p_\theta\left(Y|Z\right) = \prod_{i=1}^{N} p_\theta\left(Y_i|Y_{<i}, Z\right) \tag{4}$$

where Y consists of N tokens, Y_i is the i_{th} token of Y, and $Y_{<i}$ is the token sequence from Y_1 to Y_{i-1}.

2) Inference: MEM is not applied to source sentences at inference time because entity mentions, which provide crucial information during EL, should not be masked. The model is trained to avoid excessive reliance on entity mentions through *MEM*. As a result, it can fairly use all information even if mention masking is not applied during inference. *MEM* is not applied to the report during inference and so the condition in (5) is X, not Z.

$$\hat{Y_t} = \underset{Y_t \in \mathbf{V}}{argmax} \log p_\theta\left(Y_t|\hat{Y}_{<t}, X\right) \tag{5}$$

where $\mathbf{V}$ is vocabulary.

IV. Experiment

A. Experimental Setup

The dataset utilized in this study comprises unstructured battlefield reports containing content related to the battlefield situation. Accessing actual battlefield reports distributed within the military is restricted for security reasons. Furthermore, most countries have not recently experienced war for an extended period. True battlefield situation records are scarce. We consequently implemented a three-step process to generate data that closely resembles actual battlefield situation data. First, we engaged Korean military experts to design military operation scenarios for the Army, Navy, and Air Force. Second, we employed the Army, Navy, and Air Force simulation models used by the Korean military during actual military training. These simulation models were developed to replicate various battlefield situations based on predefined scenarios. Third, we generated a simulated battlefield report based on the information produced by the simulation models that mirrors the actual battlefield report generation process. Military experts were involved in each of these steps. The resulting simulated reports include details such as time, location, speed, direction, altitude, and quantity that effectively reflect actual battlefield situations faced by Army, Navy, and Air Force troops. Example reports are illustrated in Table I. The dataset created through this series of steps comprises 53,896 instances that encompass a total of 963 entities.

The evaluation approach was stringent. If the model accurately generated both entity mentions and entity names, it was deemed a "correct answer." Conversely, if it failed to generate either correctly, it was categorized as a "wrong answer." For training and evaluation, we randomly divided the dataset into training, development, and test sets with a ratio of 8:1:1.

B. Implementation Details

We implemented our model using PyTorch. Our experiments utilized a pre-trained T5-base [14] model and set the batch size to 16. Training employed the Adamw optimizer [18] with an initial learning rate of 5e-05, beta1 of 0.9, beta2 of 0.999, and epsilon of 1e-08. We employed teacher forcing [19] during training, where the target token is provided as the next input to the decoder. A different approach was used at inference time such that the model's predicted token, rather than the target token, is passed to the decoder as the next input. Beam search [20] was subsequently used to output the sequence with the highest overall probability among the predicted sequences as the model's final prediction result.

C. Ablation over different Masking Thresholds

We introduced *MEM* to prevent overfitting to entity mentions and allow deeper interaction with instance-specific information. In this section, we report the test set performance based on changes in the masking threshold. The results are presented in Fig. 3.

The masking threshold refers to the probability that entity mentions are masked in a sentence. The description of various masking thresholds listed in Fig. 3 is as follows.

TABLE I
EXAMPLES OF SIMULATED REPORTS ON BATTLEFIELD SITUATIONS FOR THE ARMY, NAVY, AND AIR FORCE

Army	Navy	Air Force
At 01:49 on April 5, 2019, in the Kaesong (44.883210N, 126.723502E), our A Division GOP 2nd Battalion suffered 95 deaths, 56 injuries, 8 pieces of equipment completely destroyed, and 5 pieces of equipment half destroyed by fire from the enemy artillery battalion.	At 06:24 on February 28, 2022, six enemy landing ships were detected by a high-speed boat. The observed vessel is moving southeast at 12KTS in the sea west of Eunyul (38.511668N, 124.741945E) and is believed to be preparing to infiltrate the South Korean region.	A total of 6 Prache-1Ts were detected between 09:39 on April 12, 2022 and 09:55 on April 12, 2022. The drone flew at an average of 420 knots over the southern part of Dancheon (40.161467N, 129.051320E), north of Gangneung (38.699581N, 128.964283E), and the Oho area (38.313529N, 128.710173E) for the purpose of reconnaissance of major core facilities.

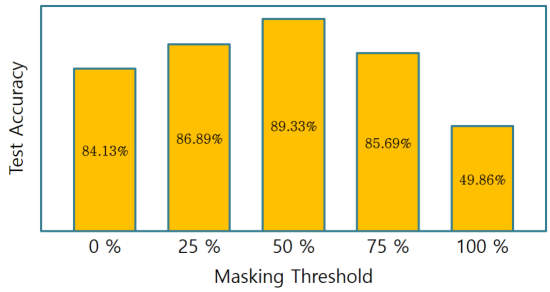

Fig. 3. Test set performance with varying masking threshold values.

- **0%:** No masking was performed for entity mentions within the sentence. Since there are no constraints on the model, the model may overly focus on features that reduce the loss of the training set the most.
- **50%:** Entity mentions in the sentence are replaced with the $[MASK]$ token with a 50% probability. During training, the model sometimes needs to make correct predictions without relying on entity mentions.
- **100%:** Masking all entity mentions in sentences. During training, the model cannot take any hints from entity mentions.

We observed changes in test set performance as the masking threshold changed, and the results are recorded in Fig. 3. The results clearly show that the best test set performance was recorded when the masking threshold was 50%. This ratio was ultimately utlized in *imEL*.

- **0% $\sim$ 50%:** In the 0% - 50% range, performance on unseen data improves linearly as the masking ratio for entity mentions increases.
- **50% $\sim$ 100%:** In this range, as the masking ratio for entity mentions increases, test set performance decreases because the model has fewer opportunities to learn entity mentions. As expected, there is a sharp performance drop when the masking threshold is 100%.
- **100%:** The model performs worst when all entity mentions within sentences are masked. This shows that the baseline model can be trained to rely excessively on entity mentions.

D. Effectiveness of imEL

Three methods were used in the experiment to confirm the effectiveness of the *imEL* model. Descriptions of each method are provided below. The results are shown in Table II.

- **Baseline (Instance-level EL Model):** The baseline in this experiment involves training a pre-trained T5-base model. As detailed in Section III-A, this is a sequence-to-sequence backbone model. The model is then used to generate a target sequence of entity mentions and entity names when a source sentence is input to the model. In other words, the baseline omits the *MEM* step, represented by (a) in Fig. 2.
- **Baseline without Translation Ability:** Section III-A further highlighted the advantage of treating the EL task as a translation task for capturing interactions between input and entity names. We therefore separately assessed the translation ability of the baseline model by replacing each unique entity name with a distinct atomic label, such as an arbitrary unique number. The model was then trained on this altered data set. The result is that the replaced target entity name cannot be inferred from the source sentence. This renders the model's translation ability ineffective.
- ***imEL* (Instance-level Masked EL model):** In this method, we intentionally introduced a more challenging training environment by masking crucial entity mentions that are primary sources of information for solving the EL task. This approach, similar to the effect of Dropout [16], enables the model to gain a more accurate understanding of instance-specific information. By rendering the presence of entity mentions unreliable during training, the model learns to be less reliant on them at test time. This is true even when the mentions are not masked within the input sentence.

Table II, shows a significant performance difference between "Baseline" and "Baseline without Translation Ability". This follows as the instance-level EL task is also a type of language translation task. The results clearly confirm that the translation ability of a sequence-to-sequence model is a desirable characteristic when performing EL.

Comparing the performance of "Baseline" and *imEL*, the

TABLE II
PERFORMANCE COMPARISON ON THE TRAINING SET, DEVELOPMENT SET, AND TEST SET.

Methods	Train Acc	Dev Acc	Test Acc
Baseline	86.12%	84.97%	84.13%
Baseline w/o Translation Ability	52.56%	49.32%	48.83%
imEL	85.86%	89.51%	89.33%

performance of "Baseline" is slightly higher in the training set. "Baseline" can utilize all entity mentions in the training set. *imEL*, on the other hand, utilizes a entity mention masking threshold of 50% as detailed in Section IV-C. More importantly, *imEL* performs approximately 5% better on the development set and test set. This result indicates that the model tends to overfit entity mentions when no regularization techniques are applied. Another noteworthy observation is that all samples correctly predicted by "Baseline" in the test set were also correctly predicted by *imEL*. This demonstrates that *MEM* is a technique that improves generalization performance without diminishing the capabilities of the baseline model.

V. CONCLUSION

This paper introduced the concept of building instance-level EL models that allow end users to interact with instance-specific information in a data set with many similar entities. This is a clear new research direction in EL systems that represents a significant contribution in its own right. One particular instance-level EL approach, *imEL*, was detailed and evaluated. *imEL* utilizes an entity masking methodology called *MEM* that enables PLM to gain a deeper understanding of instance-specific information by eliminating the reliance on entity mentions. *imEL* was shown to perform well in the context of military battlefield reports in particular, though the model can be applied equally well to any data set based on unstructured data containing many distinct instances of simliar entities.

Our future research includes a plan to conduct a more in-depth analysis of the characteristics of instance-specific information and to develop advanced techniques to enhance how PLM effectively learns this information. For example, one promising avenue is exploring methodologies that enable PLM to respond sensitively to changes in instance-specific information. The goal is to foster a detailed understanding of subtle differences between instances. Another aspect worth exploring is to investigate how these methodologies synergize with *MEM* to further enhance performance.

ACKNOWLEDGMENT

This work was supported by Defense Acquisition Program Administration and Defense Rapid Acquisition Technology Research Institute under the contract UC200018D, Korea.

REFERENCES

[1] A. Perevalov, D. Diefenbach, R. Usbeck, and A. Both, "Qald-9-plus: A multilingual dataset for question answering over dbpedia and wikidata translated by native speakers," in *2022 IEEE 16th International Conference on Semantic Computing (ICSC)*. IEEE, 2022, pp. 229–234.

[2] R. Das, M. Zaheer, D. Thai, A. Godbole, E. Perez, J. Y. Lee, L. Tan, L. Polymenakos, and A. McCallum, "Case-based reasoning for natural language queries over knowledge bases," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 9594–9611.

[3] L. Logeswaran, M.-W. Chang, K. Lee, K. Toutanova, J. Devlin, and H. Lee, "Zero-shot entity linking by reading entity descriptions," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 3449–3460.

[4] L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer, "Scalable zero-shot entity linking with dense entity retrieval," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6397–6407.

[5] Y. Onoe and G. Durrett, "Fine-grained entity typing for domain independent entity linking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 8576–8583.

[6] J. Raiman and O. Raiman, "Deeptype: multilingual entity linking by neural type system evolution," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[7] N. De Cao, G. Izacard, S. Riedel, and F. Petroni, "Autoregressive entity retrieval," in *ICLR 2021-9th International Conference on Learning Representations*, vol. 2021. ICLR, 2020.

[8] T. Ayoola, S. Tyagi, J. Fisher, C. Christodoulopoulos, and A. Pierleoni, "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*, 2022, pp. 209–220.

[9] T. Ayoola, J. Fisher, and A. Pierleoni, "Improving entity disambiguation by reasoning over a knowledge base," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022, pp. 2899–2912.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[11] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[12] H. Chang, H. Xu, J. van Genabith, D. Xiong, and H. Zan, "Joiner-bart: Joint entity and relation extraction with constrained decoding, representation reuse and fusion," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[13] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[14] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[15] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[17] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 1096–1103.

[18] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.

[19] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[20] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.