KIQA: Knowledge-Infused Question Answering Model for Financial Table-Text Data

Rungsiman Nararatwong[1], Natthawut Kertkeidkachorn[2], Ryutaro Ichise[4,1,3]
[1]National Institute of Advanced Science and Technology, Japan
[2]Japan Advanced Institute of Science and Technology
[3]National Institute of Informatics, Japan
[4]Tokyo Institute of Technology
r.nararatwong@aist.go.jp, natt@jaist.ac.jp
ichise@iee.e.titech.ac.jp

Abstract

While entity retrieval models continue to advance their capabilities, our understanding of their wide-ranging applications is limited, especially in domain-specific settings. We highlighted this issue by using recent general-domain entity-linking models, LUKE and GENRE, to inject external knowledge into a question-answering (QA) model for a financial QA task with a hybrid tabular-textual dataset. We found that both models improved the baseline model by 1.57% overall and 8.86% on textual data. Nonetheless, the challenge remains as they still struggle to handle tabular inputs. We subsequently conducted a comprehensive attention-weight analysis, revealing how LUKE utilizes external knowledge supplied by GENRE. The analysis also elaborates how the injection of symbolic knowledge can be helpful and what needs further improvement, paving the way for future research on this challenging QA task and advancing our understanding of how a language model incorporates external knowledge.

1 Introduction

Decades of development in question-answering research have seen numerous methods focusing on unstructured text, structured knowledge bases, or semi-structured tables. Recent work (Zhu et al., 2021) has discovered a new challenge in applying these techniques to the financial domain. The study proposed a QA task on financial reports compiled as a Tabular And Textual dataset for Question Answering (TAT-QA). Each question has an associated table and multiple paragraphs, making a hybrid data structure. TAT-QA requires a certain level of financial knowledge to extract evidence from tables and texts, making it an appropriate choice for our study. Our motivation is to examine whether injecting symbolic knowledge help the model better understand financial concepts.

As shown in Figure 1, we can inject the entity information of companies (dbpedia:BCE_Inc), financial terms (dbpedia:Share_repurchase), and common knowledge (dbpedia:Europe), among others. The coverage and accuracy of the information depend on the entity linking method. Nevertheless, we expect certain common entities to appear in a text-question or table-question pair. We hypothesized that this commonality helps the QA model to focus on the target answer spans, and our analysis provided evidence to confirm the hypothesis.

In summary, we introduced the knowledge-infused question answering (KIQA) model for tabular-textual data. We designed our experiment to evaluate the end-to-end results and investigate the strengths and weaknesses of the injection method to provide insights for future research. Our main contributions are as follows:

- We proposed, evaluated, and compared KIQA in different settings, improving the performance of the baseline method.
- We conducted an exhaustive attention-weight analysis of the entity-linking model we applied to our study.

Our analysis aims at understanding how language models utilize symbolic knowledge. We intend for this work to stimulate more studies into the mechanism of these models as we advance their capabilities and applications.

2 Related Works

2.1 Question Answering

Numerous QA datasets focus on textual data, such as SQuAD (Rajpurkar et al., 2016), tabular data, such as SQA (Iyyer et al., 2017) and a mixture of tables and texts (Chen et al., 2020). TAT-QA combines both tabular and textual input and requires numerical reasoning. We are interested in TAT-QA due to its practical applications since it consists of

Proceedings of Deep Learning Inside Out (DeeLIO 2022):
The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pages 53 - 61

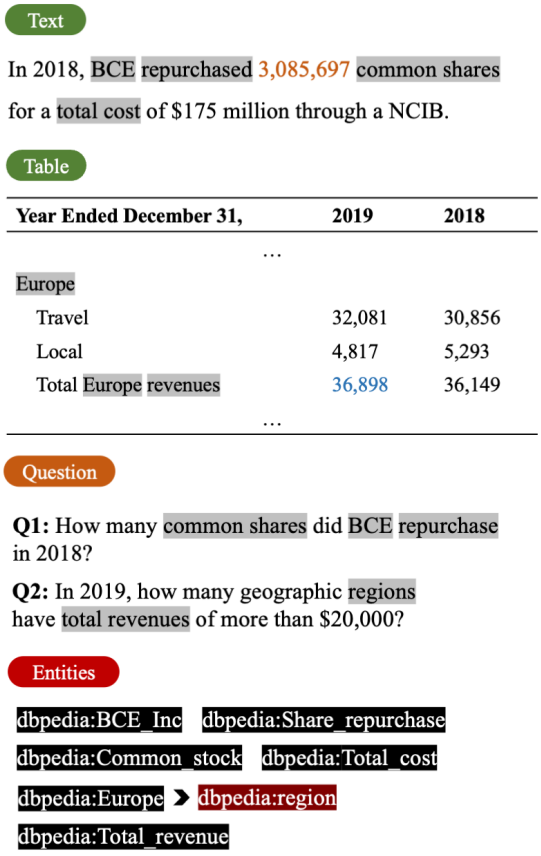

Figure 1: KIQA injects entity information commonly found in TAT-QA's tables, texts, and questions into the QA model. Some questions may require external knowledge to reason. For example, to answer Q2, the model needs to understand which cells in the table refer to a region.

real-world financial reports annotated and verified by experts. It also requires the model to understand financial concepts, making it suitable for our purposes.

TAT-QA proposed a baseline model called TagOp, which performs sequence tagging and symbolic reasoning using operators. Their experiment includes baseline textual QA models, a tabular model, and a hybrid model. TagOp significantly outperformed all baseline models; thus, we decided to base our model on it.

2.2 Entity Linking

There are several entity-retrieval models currently available, e.g., BLINK (Li et al., 2020), EntQA (Zhang et al., 2022). However, we decided to use LUKE (Yamada et al., 2020), a pre-trained language model with entity-aware self-attention, since it outputs contextualized representations of words and entities, which we can adapt to TagOp's architecture. LUKE, adapting RoBERTa's architecture (Liu et al., 2019), consists of a modified multi-layer bidirectional transformer that takes words and entities as input tokens. The modified transformer adds query matrices that allow the *entity-aware* attention mechanism to attend to both words and entities as it computes the attention scores. This additional calculation allows LUKE to directly model the relationships among words and entities.

While masked entities are part of LUKE's pretraining data, its experiment showed that explicitly adding entity information to the model's input yielded the best result. Thus, we used the GENRE (Generative ENtity REtrieval) (Cao et al., 2021) model to retrieve entities in TAT-QA and input the additional information to LUKE. Based on a pre-trained language model BART (Lewis et al., 2020), GENRE retrieves entities by generating their unique names autoregressively using constrained beam search. Given an input text sequence, the model outputs the same sequence with special tokens indicating mentions, followed by the entity's unique Wikipedia page title after each mention. For example, an output for "*In 2018, BCE repurchased 3,085,697 ...,*" is "*In 2018, **[BCE]**(BCE_Inc) **[repurchased]**(Share_repurchase) 3,085,697 ...*"

3 KIQA Model

KIQA is a QA model built from TagOp, a baseline model for the TAT-QA dataset, to evaluate symbolic knowledge injection into a QA model for a domain-specific dataset with tabular and textual structure. With the stated objective, we strictly applied the architecture of TagOp but replaced the underlying LM, RoBERTa, with LUKE to obtain knowledge-infused representations. Following TagOp, KIQA consists of three main components: 1) Evidence Extraction, 2) Reasoning and 3)Knowledge Injection.

3.1 Evidence Extraction

The evidence extraction module predicts answer spans using sequential Inside-Outside (IO) tagging Ramshaw and Marcus, 1999. TagOp takes in an input sequence of the question, flattened table, and relevant paragraphs. The preprocessing step concatenates all table cell tokens into a continuous string without separating tokens. We split KIQA into two modules, shown in Figure 2; the first module ($\text{KIQA}^{\text{TagOP}}$) is identical to TagOp, while the second module ($\text{KIQA}^{\text{Text}}$) only processes the ques-

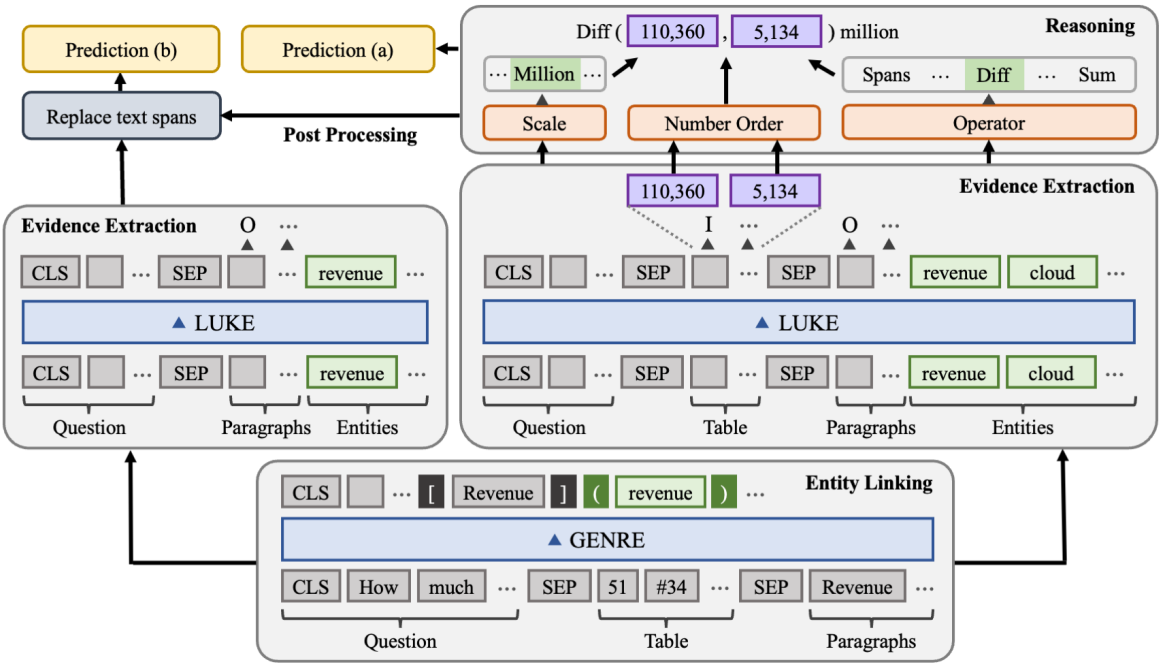

Figure 2: KIQA adopts TagOp's architecture with additional modules to handle knowledge injection. We used GENRE to retrieve entities (bottom block) and then extracted answer spans using LUKE (middle blocks). The model performs reasoning (upper-right block) on the hybrid answer spans (middle-right block). These two blocks on the right side without entity injection are comparable to TagOp. We replaced the reasoner's text span predictions with outputs from the text-only extractor (middle-left block) in certain experimental conditions.

tion and paragraphs. Our decision to introduce (KIQA$^{\text{Text}}$) stemmed from our preliminary investigation, which indicated that LUKE and GENRE did not perform as well on the tabular data as on the textual data. The idea was to replace KIQA $^{\text{TagOp}}$'s prediction on textual input with KIQA$^{\text{Text}}$'s output and measure the difference. Although the inputs are different, we applied the same two-layer feed-forward network (FFN) with GELU Hendrycks and Gimpel, 2016 activation for tag prediction:

$$\mathbf{p}_{\text{t}}^{\text{tag}} = \text{softmax}(\text{FFN}(h_t)) \quad (1)$$

where h_t is the representation of sub-token t.

3.2 Reasoning

Reasoning in TAT-QA's context involves identifying and applying an operation, such as arithmetic calculation, to the tagged sequence. Three TAGOP's components perform symbolic reasoning: operator, number order, and scale classifiers. All three classifiers are two-layer feed-forward networks with GELU activation. TagOp defines ten operators: *span-in-text*, *cell-in-table*, *spans*, *sum*, *count*, *average*, *multiplication*, *division*, *difference*, and *change ratio*. Following our early investigation, we decided to merge span-based prediction, i.e., KIQA outputs all predicted answer spans when it predicts the operator as *span-in-text*, *cell-in-table*, or *spans*. The number order classifier determines the positions of two tokens with the highest probability for *division*, *difference*, and *change-ratio* operations (e.g., the numerator and denominator in the case of division). Lastly, the scale classifier can output *none*, *thousand*, *million*, *billion*, or *percent*. Since KIQA$^{\text{Text}}$ only performs sequence tagging, it does not require the reasoning classifiers. To clarify, following TagOp's definitions,

$$\mathbf{p}^{\text{op}} = \text{softmax}(\text{FFN}([\text{CLS}])) \quad (2)$$

$$\mathbf{p}^{\text{order}} = \text{softmax}(\text{FFN}(\text{avg}(h_{t1}, h_{t2}))) \quad (3)$$

$$\mathbf{p}^{\text{scale}} = \text{softmax}(\text{FFN}([[\text{CLS}]; h_{tab}; h_p])) \quad (4)$$

where [CLS] is a sentence-level classification token, "avg" is averaging, h_{t1}, h_{t2}, h_{tab}, and h_p are the output representations of the top two tokens and the averaged representations of the table and paragraphs respectively.

3.3 Knowledge Injection

We injected symbolic knowledge to TagOp by introducing entity information obtained from GENRE to

LUKE. LUKE's transformer-based architecture allows us to fine-tune the model on downstream tasks such as QA. However, while the model learned to utilize symbolic knowledge from pre-training, it still needs additional entity information to maximize its performance (more detail in the discussion section). We obtained this information from GENRE (Cao et al., 2021). The entity retrieval model outputs unique entities' Wikipedia page titles, which we mapped to LUKE's entity vocabulary. We could map 76.92% of entities in the questions identified by GENRE to LUKE's vocabulary, averaging 1.78 entities per question. The coverage is 78.42% (0.62 entities per cell) and 64.03% (2.91 entities per paragraph) for tables and paragraphs.

3.4 Training

We trained KIQA$^{\text{TagOp}}$ and KIQA$^{\text{Text}}$ separately to measure the effect of the flattened tables, where the input contains minimal syntactic structure, and observe how LUKE and GENRE learn and generalize. Following TagOp, KIQA uses the sum of sequence tagging, operator, scale, and order classification losses (negative log-likelihood) in its optimization. We used the development set of TAT-QA for evaluating our fine-tuning to ensure consistency.

4 Experiments and Results

Our experimental settings aim to measure the effect of injecting symbolic knowledge into a domain-specific tabular/textual QA model. We chose the financial domain for evaluation since research involving knowledge-infused language models in this domain is still limited. As for the dataset, TAT-QA provides extensive and high-quality samples with complex and realistic tabular and textual data.

4.1 Dataset

TAT-QA (Tabular And Textual dataset for Question Answering) presents the challenges of performing QA on tabular/textual financial reports. The dataset consists of 16,552 questions with 2,757 hybrid contexts from 182 financial documents. Each sample contains a question, a table with $3 \sim 30$ rows and $3 \sim 6$ columns, and a minimum of two relevant paragraphs. Also included in the sample are the answer and derivation, which explain the calculation steps required to derive the answer. TAT-QA splits into three parts, i.e., training (80%), development (10%), and testing (10%). The labels in the test set are not publicly available.

Group	TagOp-based Models	Text Span Replacement
I	RB, L, L&G	-
II	RB	RB $\rightarrow$ RB
	L	L $\rightarrow$ L
	L&G	L&G $\rightarrow$ L&G
III	RB	RB $\rightarrow$ L
		RB $\rightarrow$ L&G

Table 1: The TagOp-based models make prediction on both tabular and textual data. In group II and III, we replace the hybrid models' text span predictions with text-only models' outputs (indicated by $\rightarrow$). RB = RoBERTa, L = LUKE, L&G = LUKE & GENRE.

4.2 Pipelines

We defined three groups of pipelines, each containing an ensemble of the three models we investigated. The first group includes three pipelines evaluating RoBERTa, LUKE, and LUKE with the extra entity information from GENRE (L&G). The second group replaces KIQA$^{\text{TagOp}}$'s answer span predictions from the first group with their corresponding KIQA$^{\text{Text}}$'s predictions for the *span-in-text* operator. Specifically, we replaced KIQA$^{\text{TagOp}}_{\text{RoBERTa}}$ with KIQA$^{\text{Text}}_{\text{RoBERTa}}$ and the same for LUKE and L&G. The third group is a follow-up experiment based on our analysis of the results from the first and second groups. In this last group, we paired KIQA$^{\text{TagOp}}_{\text{RoBERTa}}$ with KIQA$^{\text{Text}}_{\text{LUKE}}$ and KIQA$^{\text{Text}}_{\text{L\&G}}$ individually. We summarized our pipelines in Table 1.

4.3 Data Preprocessing

TagOp uses an automated approach to create labels for sequence tagging. We found that their algorithm does not always produce correct labeling. Therefore, we performed a simple check by extracting answer spans indicated in the labels, then executed the operations and compared the predicted answers with gold answers. Once we had identified the discrepancies, we manually examined and corrected them. However, due to the design of TagOp, we could not fix all the errors. For example, TagOp considers a table cell as a word, but some answers do not cover the entire cell. Nevertheless, since most labels are already valid, we have decided not to pursue further correction for this study. The strategy we employed was to train our models with correct samples, then validate and test the models with the entire development and test sets.

4.4 Evaluation

Table 2 shows the test set's results. The first row, TagOp, is the scores reported in the TAT-QA paper. The first pipeline of group I, RB or RoBERTa, is our reimplementation of TagOp. We attribute the boost from the original implementation to our data preprocessing algorithm, including labeling correction and elimination of invalid samples. The change we made to the prediction, i.e., outputting all answer spans for *span-in-text*, *cell-in-table*, and *spans*, also contributed to the improvement.

Although the changes we made helped increase the model's performance, it appeared that injecting external knowledge did not lead to further overall improvement. More importantly, RoBERTa seemed to outperform LUKE and GENRE on tabular data (table and hybrid). However, we noticed that LUKE & GENRE consistently exceeds RoBERTa in arithmetic operations and single-span prediction. While the arithmetic score results from multi-step prediction involving reasoning, single-span answers are more straightforward to isolate and measure the effect of knowledge infusion.

Based on group I and II results, we created the third group of pipelines consisting of $\text{KIQA}^{\text{TagOp}}_{\text{RoBERTa}}$ paring with the text-based models $\text{KIQA}^{\text{Text}}_{\text{RoBERTa}}$, $\text{KIQA}^{\text{TagOp}}_{\text{Luke}}$, and $\text{KIQA}^{\text{Text}}_{\text{L\&G}}$. The results indicate that injecting external knowledge into the textual part of the data improves the QA model. Nonetheless, due to the hybrid nature of the dataset, the overall improvement is less dramatic. According to our analysis of the training data, it is likely that the high variance in the counting columns is due to the small number of samples in this category.

5 Analysis

We have learned from our experimental results that injecting entity information helped improve the model's performance on textual data. This conclusion seems reasonable given that we did not provide the model with the same information for the tabular input. However, in some cases, the infused text-only entity information negatively affects the model's ability to handle tabular input. Our analysis attempts to answer the following questions:

- **Q1:** How does the injected external knowledge contribute to the improvement?
- **Q2:** Why do the knowledge-infused models underperform the baseline model on tabular data?

5.1 Attention Weights

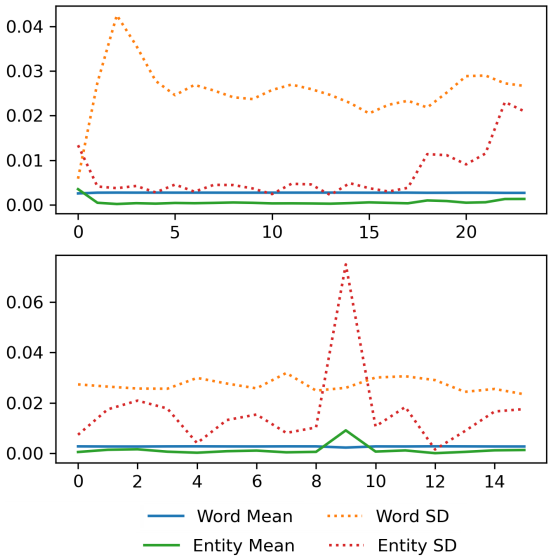

Figure 3: Top: Average and standard deviation of attention scores by layer (0 ~23). Bottom: Average and standard deviation of layer 22's attention scores by attention head (0 ~15).

We investigated Transformers' (Vaswani et al., 2017) attention weights α in different levels of aggregation to determine how LUKE utilizes entity information. Each Transformers layer consists of multiple attention heads. LUKE employs entity-aware self-attention, meaning that the model computes the weights from both word and entity tokens:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \alpha \mathbf{V} \tag{5}$$

$$\alpha = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^{\text{T}}}{\sqrt{L}}) \tag{6}$$

where the query matrix $\mathbf{Q} \in \mathbb{R}^{L \times D}$ can be one of $\mathbf{Q}_{w2w}$, $\mathbf{Q}_{w2e}$, $\mathbf{Q}_{e2w}$, or $\mathbf{Q}_{e2e}$, depending on the types of tokens (word or entity). $\mathbf{K} \in \mathbb{R}^{L \times D}$ and $\mathbf{V} \in \mathbb{R}^{L \times D}$ denote key and value matrices. L is the dimension of input embedding, and D is the dimension of output embedding.

LUKE and RoBERTa (large model) consist of 24 layers of Transformers with 16 attention heads on each level. RoBERTa has an input length of 512 tokens, while LUKE extends it to 549 to handle the entity input, resulting in up to 549 $\times$ 549 attention matrix. Taken together with the 1,668 samples in TAT-QA's development set, the analysis would involve 12-billion data points.

The most straightforward approach is to average the weights by layer, head, and sample. However,

Model	EM	F1	Table A	Table C	Table M	Table S	Hybrid A	Hybrid C	Hybrid M	Hybrid S	Text A	Text M	Text S
TagOp	50.1	58.0	41.1	63.6	66.3	56.5	46.5	62.1	**63.2**	68.2	27.3	19.0	45.2
Group I: TagOp-based Models													
RB	57.2	67.2	51.6	36.4	**72.3**	**60.7**	**63.3**	**79.3**	60.4	68.8	18.2	19.1	51.1
LUKE	54.3	64.8	47.4	**72.7**	65.1	57.8	48.9	62.1	**62.3**	**75.5**	27.3	14.3	51.1
L&G	56.4	66.4	**53.7**	27.3	65.1	59.0	55.4	51.7	58.5	72.9	27.3	19.1	52.3
Group II: TagOp-based & Text Models													
RB	57.3	67.2	51.6	63.7	68.7	58.3	62.3	65.5	**62.3**	73.4	27.3	19.1	50.8
LUKE	56.4	66.1	52.8	45.5	68.7	55.4	58.6	34.5	61.3	75.0	27.3	19.1	51.1
L&G	57.2	66.6	**53.7**	27.3	65.1	59.5	55.4	51.7	58.5	75.0	27.3	19.1	54.8
Group III: TagOp-based (RB) & Text Models (LUKE & L&G)													
RB	57.3	67.2	51.6	63.7	68.7	58.3	62.3	65.5	**62.3**	73.4	27.3	19.1	50.8
LUKE*	57.6	67.1	•	•	•	59.0	•	•	•	74.5	•	•	51.7
L&G*	**58.2**	**67.4**	•	•	•	59.0	•	•	•	73.0	•	•	**55.3**

Table 2: Evaluation of the first and second groups on the test set. The abbreviations are: RB = RoBERTa, A = Arithmetic, C = Counting, M = Multi-span extraction, S = Single-span extraction. The detailed scores are exact match (EM) scores. The underlined scores are the top scores in the group, and the top scores across all groups are in bold. The test set does not include samples with the counting operation, so we removed them from the table. * For group III, since we only replaced RoBERTa's text span outputs with the text-only LUKE and L&G's outputs, the scores for A, C, and M are the same as those of RoBERTa (indicated by •).

since most tokens are unrelated to the entities, averaging the entire input sequence would dampen any indication of high attention paid to the entities. We instead narrowed our focus to tokens within the correct answer spans. In other words, where does the model pay attention when it computes output representations of the answer tokens?

Given an input sequence $\mathbf{x} = (x_1, ..., x_n)$ and a target output $\mathbf{y} = (y_1, ..., y_n) \in \{0, 1\}$, where $y_i = 1$ if y_i belongs to an answer span $s \in S$, let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be an attention-weight matrix. We selected $\mathbf{a}_i \in \mathbf{A}$ where $y_i = 1$ to form a reduced matrix $\tilde{\mathbf{A}} \in \mathbb{R}^{k \times n}$, then averaged $\tilde{\mathbf{A}}$ along the first dimension to produce vector $\mathbf{b}$, representing averaged attention weights of the answer tokens. Since we are interested in all m samples individually, we based our analysis on matrix $\mathbf{B} = [\mathbf{b}_1, ..., \mathbf{b}_m]$, where:

$$\mathbf{b} = \frac{\sum_{j=1}^{k} \tilde{\mathbf{a}}_j}{k}, \tilde{\mathbf{A}} = [\tilde{\mathbf{a}}_1, ..., \tilde{\mathbf{a}}_n] \quad (7)$$

First, we looked for the layers where the model pays heightened attention to the entities. We obtained this information by averaging $\mathbf{B}$ over each layer's attention heads, as shown in Figure 3 (a). Interestingly, the standard deviations indicate that the model pays special attention to the entities on its top layers. In Figure 3 (b), we took a closer look at layer 22 and found that attention head 9 seemed to specialize in the infused knowledge. We observed a similar trend on layer 23 but chose to analyze layer 22 as its standard deviation was the highest among all layers.

5.2 Visualizing Attention

Figure 4 shows averaged attention weights by sample. We sorted the samples by their maximum attention score among the entities since the model tends to pay attention to specific entities rather than all of them when computing the representations of the target tokens. We refer to these maximum scores as *relevance* scores. Since RoBERTa does not have entity inputs, we sorted the samples based on LUKE's scores. While the sequence lengths are varied, they all start with the sentence-level classification token, followed by the question, flattened table, and paragraphs. LUKE has additional attention weights starting from b_{513} to b_{549}. We included Figure 6 as a reference for tabular and textual input boundaries.

Since we only injected entity information to the textual part of the data, it is reasonable that the model would pay more attention to the entities for samples where the answer spans are in paragraphs.

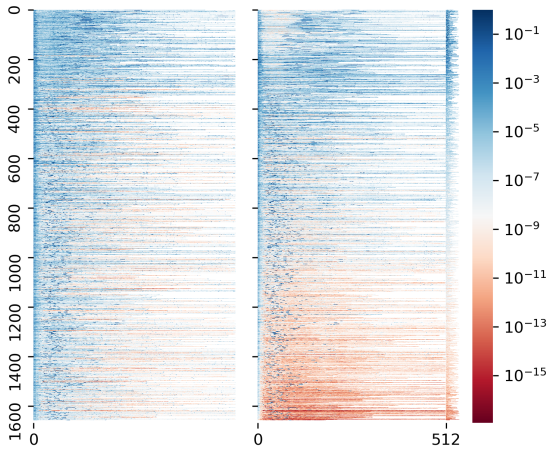

Figure 4: Attention weights of samples sorted by the relevance scores, the maximum attention scores among entities. The left side (a) is a heat map in log scale for $\text{KIQA}^{\text{TagOp}}_{\text{RoBERTa}}$ and the right side (b) is for $\text{KIQA}^{\text{TagOp}}_{\text{L\&G}}$.

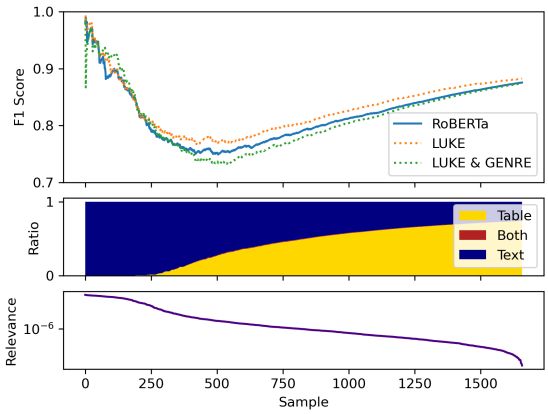

Figure 5: From top to bottom: (a) The average of accumulated F1 scores sorted by the relevance score, (b) the accumulated ratio of answer span locations in the input sequences, (c) the relevance score (in log scale) computed from the maximum attention weight among entities.

This pattern is most visible in Figure 6 (a), where the entity's attention weights decrease as the model attends more to the tabular part.

In Figure 4, we observed a pattern of difference in attention weights among samples where LUKE pays more attention to the entities. While the answer spans in these samples are in the paragraphs, RoBERTa seems to pay considerable attention to the tabular inputs. On the other hand, LUKE seems more focused on the textual part. This pattern clearly shows that the infused knowledge helps guide the model to narrow its focus to the more relevant section. While we did not observe the opposite effect since we did not inject entity information into the tabular part, with an entity retrieval model capable of linking tabular data, there is a possibility that the model may behave as expected. Nevertheless, this observation warrants further study on integrating entity retrieval models specializing in tabular data.

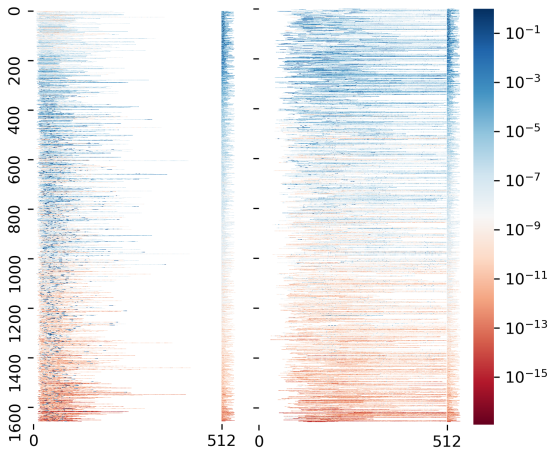

Figure 6: Table and paragraph boundaries in terms of attention weights. The left side (a) includes the scores of the sentence-level classification tokens, questions, and flattened tables. The right side (b) is the paragraph tokens' scores.

5.3 Interpretation

We learned from the previous section that the entity information helps guide the model to pay attention to the more relevant part of the input. The next and crucial question is whether or not this change of focus translates into improved accuracy. We used the F1 score that exclusively measures sequence-tagging prediction and omitted the reasoning operations to isolate the effect of knowledge infusion. Our objective is to find patterns in the model's performance (Figure 5) in relation to how the model utilizes the entity information (Figure 4) that could explain the two questions we posed at the beginning of the analysis.

We created the accumulated F1-score chart in Figure 5 (a) based on the sorted attention weight vectors as in the heat map in Figure 2. To clarify, the score at the i^{th} position on the x-axis is the average of F1 scores from the first sample to the i^{th} sample. The corresponding ratio chart (b) is also an accumulated ratio of the same sequence of samples, i.e., the ratio of text and tabular-based questions in the top-i^{th} samples. However, the relevance score is of the individual sample at the i^{th} position.

The F1-score chart exhibits different patterns at different sample ranges; therefore, we divided our interpretation into four parts. The first part

starts from the first sample to roughly the 50^{th} sample. While the F1 scores within this range are high, their margin is minuscule, indicating that the questions are relatively easy enough that the baseline language model can predict the correct answers without help from the infused knowledge.

The second part (approximately 50^{th}∼200^{th}) is where LUKE & GENRE has the most advantage. The rapid drop in the F1 scores across all models means that the text-based questions are much more difficult. The exact section in Figure 4 shows that the infused knowledge is still highly relevant in directing the model's attention until this point. We sampled question-answer pairs with the entity and attention information from this part and will discuss them in the following section.

The majority of the samples in the third part (200^{th}∼1000^{th}) are table-based questions, as indicated by the steady increase in their ratio. According to Figure 2, the model pays less attention to the entities than the first two parts, although still noticeably higher than the fourth part. Since the answers are in the tables but the entities link to mentions in the paragraphs, they are not particularly useful. On the contrary, the potentially unrelated information weakens LUKE's performance considerably.

The last part (1000^{th}∼1668^{th}), also primarily table-based, is easier to answer than the previous one. As the model mostly ignores entity information, LUKE & GENRE's performance recovers steadily due to less interference.

5.4 Examples

Our examples, shown in Table 3, are from the third part of our interpretation, where the injection of external knowledge contributes most to the model's performance. We only include the entity with the highest attention score and its corresponding mention in the text for each example. These examples represent some aspects of the differences the infusion made. In the first example, according to the correct answer, the margin increased because the total margin decreased slightly due to expenses growth. RoBERTa was able to correctly predict the first half of the answer span ("Excluding the effects of currency rate fluctuations, our cloud and license segment's total"), which does not include the primary point. The entity "Expense" seems to highlight the relevance of the latter half, resulting in LUKE's complete prediction.

The second example is a precise instance of the more concentrated attention weights pattern we observed in Figure 4. Although this seems to be a complex case since no model could achieve a high score, LUKE could partially predict the correct answer. On the other hand, we examined RoBERTa's attention scores and found that the model was paying attention to the tabular part of the input.

In our opinion, while GENRE provided the precise information for EBITA, it does not seem to contribute significantly to the improvement. RoBERTa already partially captured the main reason for the increase, while the mention "EBITA" only completes the beginning of the sentence (LUKE's answer: "Adjusted EBITA was on the prior-year level as ... [main reason]"). Nonetheless, LUKE also included the entire reason while RoBERTa missed part of it, thus achieving a much better score on this sample.

In the last example, while RoBERTa correctly located the correct answer span, it also included irrelevant adjacent text, negatively affecting the F1 score considerably.

Q-113: Why did the cloud license segments total margin increase ...? **Mention:** ... due to expenses growth. **Entity:** Expense **F1 scores:** L&G = 1.00, RB = 0.54
Q-139: When is the impairment of goodwill and tangible assets tested? **Mention:** intangible assets is tested annually **Entity:** Intangible asset **F1 scores:** L&G = 0.38, RB = 0.00
Q-156: What was the reason for the increase in the Adjusted EBITDA? **Mention:** Adjusted EBITA was on the ... **Entity:** Earnings before interest, taxes, depreciation, and amortization **F1 scores:** L&G = 1.00, RB = 0.68
Q-178: When does the company record an accrued receivable? **Mention:** ... prior to invoicing ... **Entity:** Contractual term **F1 scores:** L&G = 1.00, RB = 0.39

Table 3: Example $\text{KIQA}^{\text{TagOp}}_{\text{L\&G}}$s, including the entity with maximum α and its corresponding mention.

6 Discussion

The QA model used the infused knowledge to focus on the more relevant information (**Q1**). However, only 25.20 % of the answers are in the paragraphs, explaining the limited improvement. We did not anticipate the margin to be substantial since LUKE's EM score on the development set of the SQuAD 1.1 dataset (Rajpurkar et al., 2016) was only 1.01 % (88.9 → 89.8). Injecting entity information to LUKE resulted in 0.21 % improvement (94.8 → 95.0). However, since our baseline score is much lower, it was reasonable to expect a higher increase (RoBERTa → LUKE & GENRE: 8.86 % for single text spans). Our analysis revealed that the irrelevant entity information interfered with the model's decision, which is why the knowledge-infused models underperformed the baseline model (**Q2**).

There is still a gap in TAT-QA's tabular data where GENRE did not perform well, requiring further study involving entity-linking models specialized in tabular data. Solving the problem of unrelated entity information interfering with the model's prediction is also another challenge.

7 Conclusion

We investigated the effect of external knowledge infusion on a hybrid tabular/textual QA model in the financial domain. The results indicated an improvement, especially to the textual part of the data. Our attention-weight analysis shows the model's ability to utilize the injected knowledge and reveals the challenges involving the hybrid structure of the data. As a result, this study has paved the way for future research to incorporate entity-linking models specialized in tabular data and find a solution that enables the model to integrate tabular and textual symbolic knowledge more efficiently.

References

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. 2021. Autoregressive entity retrieval. In *International Conference on Learning Representations*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Wang. 2020. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP*.

Dan Hendrycks and Kevin Gimpel. 2016. Bridging non-linearities and stochastic regularizers with gaussian error linear units. *arXiv:1606.08415*.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computation Linguistics*.

Belinda Z. Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. 2020. Efficient one-pass end-to-end entity linking for questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Wenzheng Zhang, Wenyue Hua, and Karl Stratos. 2022. EntQA: Entity linking as question answering. In *International Conference on Learning Representations*.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.