KaFSP: Knowledge-Aware Fuzzy Semantic Parsing for Conversational Question Answering over a Large-Scale Knowledge Base

Junzhuo Li[1] and **Deyi Xiong**[21] *
[1]School of New Media and Communication, Tianjin University, Tianjin, China
[2]College of Intelligence and Computing, Tianjin University, Tianjin, China
{jzli,dyxiong}@tju.edu.cn

Abstract

In this paper, we study two issues of semantic parsing approaches to conversational question answering over a large-scale knowledge base: (1) The actions defined in grammar are not sufficient to handle uncertain reasoning common in real-world scenarios. (2) Knowledge base information is not well exploited and incorporated into semantic parsing. To mitigate the two issues, we propose a knowledge-aware fuzzy semantic parsing framework (KaFSP). It defines fuzzy comparison operations in the grammar system for uncertain reasoning based on the fuzzy set theory. In order to enhance the interaction between semantic parsing and knowledge base, we incorporate entity triples from the knowledge base into a knowledge-aware entity disambiguation module. Additionally, we propose a multi-label classification framework to not only capture correlations between entity types and relations but also detect knowledge base information relevant to the current utterance. Both enhancements are based on pre-trained language models. Experiments on a large-scale conversational question answering benchmark demonstrate that the proposed KaFSP achieves significant improvements over previous state-of-the-art models, setting new SOTA results on 8 out of 10 question types, gaining improvements of over 10% F1 or accuracy on 3 question types, and improving overall F1 from 83.01% to 85.33%. The source code of KaFSP is available at https://github.com/tjunlp-lab/KaFSP.

1 Introduction

With the growing popularity of intelligent virtual assistants (e.g., Alexa, Siri, Cortana) and the availability of large-scale knowledge bases (e.g., DBPedia (Auer et al., 2007), Wikidata (Vrandečić and Krötzsch, 2014), YAGO (Rebele et al., 2016)), conversational question answering (QA) over knowledge bases (KB) has attracted broad interests. It aims to satisfy users' information needs by retrieving answers from a given knowledge graph to users' questions in a multi-turn conversational setting with a wide range of discourse phenomena (e.g., ellipsis, coreference, lexical cohesion).

While conversational QA over large-scale KBs can be realized without explicit semantic parsing (e.g., HRED-KVM (Kacupaj et al., 2021)), the majority of effort is dedicated to the exploration of contextual semantic parsers (Guo et al., 2018; Shen et al., 2019; Thirukovalluru et al., 2021; Kacupaj et al., 2021; Lan and Jiang, 2021). The semantic parsing based approaches usually project an utterance into a logical form that can be executed on a given knowledge base. Early semantic parsing method D2A (Guo et al., 2018) suffers from the stepwise error propagation issue, which is improved by MaSP (Shen et al., 2019) that jointly learns pointer-equipped semantic parsing and type-aware entity detection in a multi-task learning framework. The very recent work LASAGNE (Kacupaj et al., 2021) further enhances MaSP via a graph attention network that exploits the correlation (missing in MaSP) between entity types and relations and achieves the state-of-the-art results on the CSQA benchmark (Saha et al., 2018).

Despite the aforementioned progress, we argue that current semantic parsing approaches to conversational QA over large-scale KBs still suffer from two critical issues. First, grammar rules that form the base for the mapping of questions to logical forms, although being constantly updated in D2A, MaSP, and LASAGNE, are still not sufficient to cover all real-world situations, e.g., fuzzy inference on numbers. Consider the question "*Which nutrients can interact with approximately 89 chemical substances and drugs?*". It is difficult for existing grammar to represent "*approximately 89*". Second, the interaction between questions and knowledge base is not adequate for entity disambiguation and redundancy detection in semantic parsing. For the

[*] Corresponding author.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 461 - 473
May 22-27, 2022 ©2022 Association for Computational Linguistics

question "*Which educational institution is the alma mater of Pierre Lefebvre?*", without using relevant information from KB, it is difficult for semantic parsing to distinguish whether "*Pierre Lefebvre*" is a French military physician or a French politician as more than one persons named "*Pierre Lefebvre*" are in the knowledge base.

To address these two issues, we propose a **K**nowledge-**a**ware **F**uzzy **S**emantic **P**arsing (KaFSP) model to enhance both grammar rules and the interaction between KB and semantic parsing. Particularly, we introduce fuzzy operations into the grammar system used in previous work, enabling the system to perform uncertainty reasoning on numbers. Such updates have a significant impact on answering quantitative and comparative questions. In order to make the knowledge base well facilitate semantic parsing, we incorporate deep entity knowledge in the given knowledge base into different modules in the proposed semantic parsing framework. In the entity disambiguation module, entity triples from the knowledge base are exploited to disambiguate candidate entities. In the entity type and relation prediction module, a multi-label classification framework is proposed to capture correlations between entity types and relations and to pinpoint KB information relevant to the current utterance.

Contributions Our main contributions are as follows:

- We propose a knowledge-aware fuzzy semantic parsing framework for conversational QA over large-scale KBs, which enables the grammar system to model uncertainty reasoning based on the fuzzy set theory, and enhances the interaction between KB and semantic parsing with two knowledge-aware modules.

- Experiment results demonstrate that our proposed model achieves new state-of-the-art results on 8 out of 10 question types on the CSQA dataset (Saha et al., 2018), which is to date the largest dataset for complex conversational question answering over a large-scale knowledge base.

2 Related Work

Semantic parsing approaches have conventionally been used for knowledge base question answering (KBQA). Early efforts parse natural language questions into logical forms typically via dictionary-based parsers or similarity models (Wong and Mooney, 2007; Zettlemoyer and Collins, 2007, 2009; Kwiatkowski et al., 2011; Andreas et al., 2013; Artzi and Zettlemoyer, 2013; Reddy et al., 2014; Zhao and Huang, 2015; Dubey et al., 2016; Long et al., 2016).

Recent years have witnessed that semantic parsing has been shifted from traditional statistical models with feature engineering to neural approaches that learn continuous representations for generating logical forms (Yih et al., 2014; Jia and Liang, 2016; Xiao et al., 2016; Bao et al., 2016; Dong and Lapata, 2018, 2016; Bhutani et al., 2020; Lan and Jiang, 2020, 2021). For example, Dong and Lapata (2016) use the encoder-decoder framework equipped with a neural attention mechanism to cast semantic parsing into Seq2Seq generation.

As knowledge bases are becoming large, semantic parsing for KBQA is usually performed in a stepwise, modular framework. Guo et al. (2018) recognize entities in questions and link them to the given large-scale knowledge graph at the first stage and then learn to map the entity-linked questions into logical forms. Dong and Lapata (2018) propose a coarse-to-fine two-stage decoding method for semantic parsing, which generates a coarse sketch for a question with low-level features at the first stage and then continues to decode the final logical form based on the output of the first stage as well as the question itself.

As mentioned in Section 1, such stepwise methods are confronted with error propagation across stages (e.g., from entity linking to mapping, from coarse parse to fine parse). In order to alleviate such problem, Shen et al. (2019) and Kacupaj et al. (2021) use a multi-task learning framework to jointly learn entity detection, linking, and semantic parsing in a single model. Kacupaj et al. (2021) also use a graph attention network (Veličković et al., 2018) to explore entity type and relation information in the knowledge base.

Due to the superiority of multi-task learning for semantic parsing tailored for KBQA, our work is also based on the multi-task learning framework. However, our model is significantly different from existing works in both fuzzy grammar rules and knowledge-aware entity disambiguation together with entity type and relation prediction.

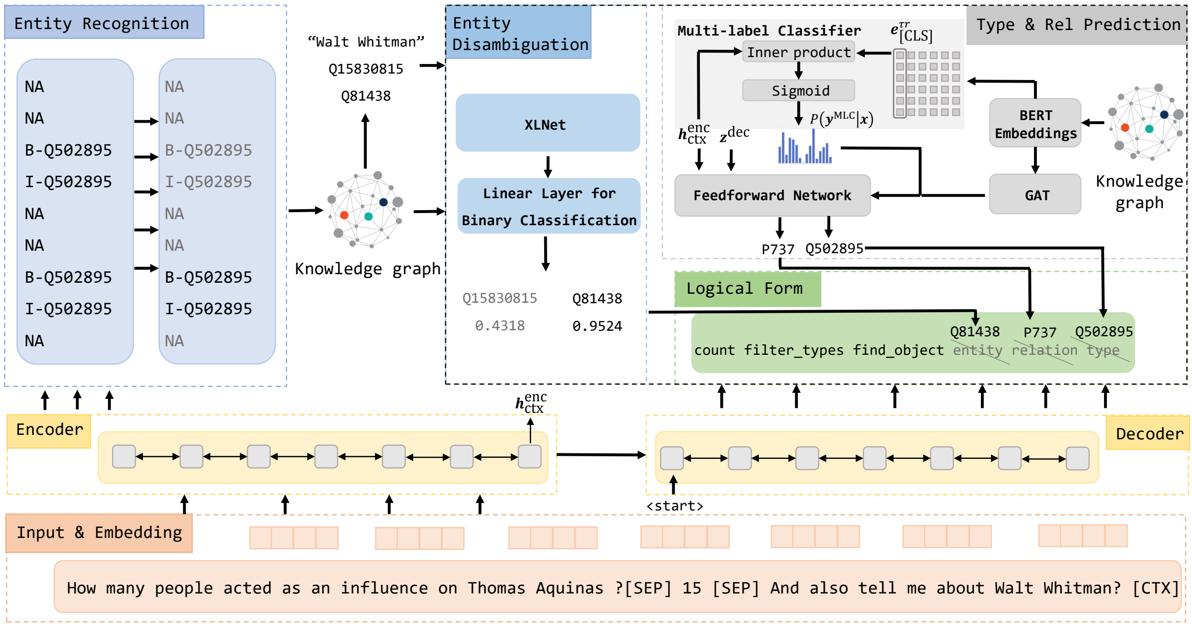

Figure 1: The diagram of KaFSP. Note that Q* and P* are entity and relation IDs defined in Wikidata where an entity type is also regarded as an entity.

3 KaFSP

We use a multi-task learning framework to map an input (current question concatenated with context) into a logical form where entities are detected and linked to the given knowledge base. Figure 1 shows the architecture of KaFSP. The backbone network of KaFSP follows LASAGNE (Kacupaj et al., 2021) consisting of a seq2seq network, an entity recognition module and a graph attention network module (Section 3.2). Our contributions lie in the fuzzy grammar (Section 3.1), the knowledge-aware entity disambiguation module (Section 3.3), and the entity type and relation prediction module (Section 3.4). The two knowledge-aware modules are shown in the black dashed box in Figure 1.

3.1 Fuzzy Grammar

In semantic parsing approaches tailored for conversational KBQA, a grammar with the minimum number of actions is usually defined to construct KB-executable logical forms (i.e., semantic parse trees). The actions defined in the previous grammar system (Guo et al., 2018; Kacupaj et al., 2021) are all deterministic operations. However, vague and fuzzy questions are common in real-world scenarios, e.g., "*How many works of art did approximately the same number of people do the dubbing for as Another ?*", which cannot be answered by previous deterministic grammars. The grammar of LASAGNE includes an action termed "approx", which aims to perform the operation of "approximately equal to". However, how two numbers are measured to be roughly equal to each other is not defined. Therefore, we take the grammar of LASAGNE as a starting point for building our own grammar and add fuzzy actions to the grammar to make it to adapt to real-world vague questions mentioned above. The new grammar is briefly summarized in Table 1.

We further give a "precise" (measurable) definition for these added fuzzy actions based on the fuzzy set theory (Zadeh, 1965). For a number a, we define its fuzzy set as $A = \{x, \mu(x)|x \in \mathbb{R}\}$. $\mu(x)$ is the membership function of set A, which indicates the degree of similarity between x and a, and is defined based on a generalized bell-shaped membership function as:

$$\mu(x) = \frac{1}{1 + |\frac{x-a}{c}|^{2b}}, \quad (1)$$

where $c \in \mathbb{R}$ and $b \in \mathbb{N}^+$. When $\mu(x) = 1$, x and a are strictly equal; and when $\mu(x) = 0$, x and a are strictly not equal.

A threshold $\lambda \in (0, 1]$ can be defined to get three

Category	Action	Description
find	set→subject(e, r)	set of subjects part of the triples with object e and relation r
	set→object(e, r)	set of objects part of the triples with subject e and relation r
	boolean→in(e, set)	check if the entity e is part of the set
filter	set→types(set$_e$, set$_t$)	filter the entity set set$_e$ based on type set set$_t$
count	num→count(set)	count the number of elements in the set
	dict→sub_triples(r, τ_1, τ_2)	extracts a dictionary, where keys are entities of type τ_1 and values are the number of objects of type τ_2 related with r
	dict→ob_ triples(r, τ_1, τ_2)	extracts a dictionary, where keys are entities of type τ_1 and values are the number of subjects of type τ_2 related with r
strict comparison	set→greater (dict, num)	set of those entities that have greater count than num
	set→lesser(dict, num)	set of those entities that have lesser count than num
	set→equal(dict, num)	set of those entities that have equal count with num
	set→atmost(dict, num)	set of those entities that have at most same count with num
	set→atleast(dict, num)	set of those entities that have at least same count with num
	set→argmax(dict, num)	set of those entities that have the most count
	set→argmin(dict, num)	set of those entities that have the least count
fuzzy comparison	set→equal(dict, num)	set of those entities that are in the fuzzy set $A_\lambda^{\approx}$ of num
	set→greater(dict, num)	set of those entities that are in the fuzzy set $A_\lambda^{\gtrsim}$ of num
	set→lesser(dict, num)	set of those entities that are in the fuzzy set $A_\lambda^{\lesssim}$ of num
set operation	set→intersection(set$_1$, set$_2$)	interesection of set$_1$ and set$_2$
	set→union(set$_1$, set$_2$)	union of set$_1$ and set$_2$
	set→difference(set$_1$, set$_2$)	difference of set$_1$ and set$_2$

Table 1: Fuzzy grammar defined for KaFSP.

fuzzy sets:

$$\begin{aligned} A_\lambda^{\approx} &= \{\mu(x) > \lambda | x \in \mathbb{R}\}, \\ A_\lambda^{\gtrsim} &= \{x > a | x \in \mathbb{R}\} \cup A_\lambda^{\approx}, \\ A_\lambda^{\lesssim} &= \{x < a | x \in \mathbb{R}\} \cup A_\lambda^{\approx}. \end{aligned} \tag{2}$$

When $\mu(x) > \lambda$, then $x \in A_\lambda^{\approx}$, which denotes that x and a is approximately equal to each other. When $x \in A_\lambda^{\gtrsim}$, x is considered to be greater than or approximately equal to a. When $x \in A_\lambda^{\lesssim}$, x is considered to be less than or approximately equal to a.

It is worth noting that all the parameters in Eq. (1) and the threshold λ can be flexibly predefined, which makes our grammar adjustable to different fuzzy scenarios.

3.2 Backbone Network

We follow the multi-task learning framework of LASAGNE (Kacupaj et al., 2021) to build the backbone network for our KaFSP.

Encoder and Decoder The skeleton of the entire model is a Transformer-based encoder-decoder network. The input $\boldsymbol{x}$ fed into the encoder is formed in a way similar to LASAGNE, which is composed of the previous question, the answer to the previous question, and the current question separated by a symbol "[SEP]". A special token "[CTX]" is appended to the input for encoding the input representation $\boldsymbol{h}_{\text{ctx}}^{\text{enc}}$, as shown in Figure 1. Both the encoder and decoder use a two-layer multi-head attention Transformer block, which can be formulated as:

$$\begin{aligned} \boldsymbol{h}^{\text{enc}} &= \text{encoder}(\boldsymbol{x}; \boldsymbol{\theta}^{\text{enc}}), \\ \boldsymbol{z}^{\text{dec}} &= \text{decoder}(\boldsymbol{h}^{\text{enc}}; \boldsymbol{\theta}^{\text{dec}}), \\ P(\boldsymbol{y}^{\text{dec}} | \boldsymbol{x}) &= \prod_t \text{softmax}(\boldsymbol{W}^{\text{dec}} \boldsymbol{z}_t^{\text{dec}}), \end{aligned} \tag{3}$$

where $\boldsymbol{z}_t^{\text{dec}} \in \mathbb{R}^{|\mathbb{V}^{\text{dec}}|}$ is the hidden state of the decoder at time step t, and $\boldsymbol{W}^{\text{dec}}$ is the linear projection matrix at the targe side. The key task of the decoder is to generate an action (listed in Table 1) at each time step to obtain the logical form $\boldsymbol{y}^{\text{dec}}$ corresponding to the input $\boldsymbol{x}$.

Entity Recognition Inspired by Shen et al. (2019), we jointly detect entities and their types in a BIO sequence labeling way. The labels for the input sequence $\boldsymbol{x}$ are in $\{O, \{B, I\} \times \{T_i\}_1^{N_{\text{tp}}}\}$. T_i stands for the i-th entity type label, and N_{tp} denotes the number of the distinct entity types in the knowledge base. An LSTM network, stacked over the encoder, is used to perform the sequence labeling task. To make the outputs of the sequence labeling task compatible with logical forms, we follow LASAGNE to use a feedforward layer stacked over the LSTM layer. The entire module of entity recognition is hence formulated as follows:

$$\begin{aligned} &\boldsymbol{h}^{\text{LSTM}} = \text{LSTM}(\boldsymbol{h}^{\text{enc}}; \boldsymbol{\theta}^{\text{LSTM}}), \\ &\boldsymbol{h}^{\text{FFN}} = \text{LeakyReLU}(\boldsymbol{W}^{\text{FFN}_1}[\boldsymbol{h}^{\text{enc}}; \boldsymbol{h}^{\text{LSTM}}]), \\ &P(\boldsymbol{y}^{\text{ER}}|\boldsymbol{x}) = \prod_t \text{softmax}(\boldsymbol{W}^{\text{FFN}_2}\boldsymbol{h}_t^{\text{FFN}}), \end{aligned} \tag{4}$$

where $\boldsymbol{h}^{\text{LSTM}}$ is the LSTM hidden state at time step t, $\boldsymbol{h}^{\text{FFN}}$ is the FFN-transformed version of $\boldsymbol{h}^{\text{LSTM}}$, and $P(\boldsymbol{y}^{\text{ER}}|\boldsymbol{x})$ denotes the probability distribution over entity tags.

Graph Attention Network (GAT) We follow LASAGNE to use the GAT module to learn the correlations between entity types and their relations in the knowledge base. It can be defined as:

$$\boldsymbol{h}^{\text{GAT}} = \text{GAT}(\boldsymbol{e}^{\text{node}}; \boldsymbol{\theta}^{\text{GAT}}), \tag{5}$$

where $\boldsymbol{e}^{\text{node}}$ are the embeddings of nodes in the type-relation graph constructed from the knowledge base. Please refer to Kacupaj et al. (2021) for more details on the GAT module.

3.3 Entity Disambiguation

In a large-scale knowledge base, it is common that entities with different meanings share the same surface forms. Predicting entity types could help differentiate them. However, when candidates have both the same type and surface form, it is difficult for entity type prediction to distinguish them again. In order to address this issue, we incorporate more information about these ambiguous entities from the knowledge base to disambiguate them.

We model the entity disambiguation problem as a binary classification problem:

$$y = f(\boldsymbol{c}, s, \mathcal{K}(e)), \tag{6}$$

where s is the surface form of a candidate entity e, $\boldsymbol{c}$ is the context where e occurs, and $\mathcal{K}(e)$ denotes relevant information of the candidate entity e from the knowledge base. If $y = 1$ the entity e is disambiguated and linked to the true entity in the knowledge base defined by $\mathcal{K}(e)$. The purpose of this is to maximize both the true positive and true negative.

We define the context of e as the entire input $\boldsymbol{x}$. To define $\mathcal{K}(e)$, we use all triples that are relevant to e in the knowledge base, regardless of whether the entity is a subject or an object in triples. That is, $\mathcal{K}(e)$ is an ordered set of KB triples. Each triple in $\mathcal{K}(e)$ can be formulated as (e_h, r, e_t), where the candidate entity e is either the head entity (e_h) or tail entity (e_t).

In Eq. (6), f is the classifier to disambiguate candidate entities. We use a pre-trained language model XLNet (Yang et al., 2019) fine-tuned in the training dataset as the classifier.

In order to feed s, $\boldsymbol{c}$, and $\mathcal{K}(e)$ into the pre-trained and fine-tuned classifier, we reorganize them into a concatenated textual sequence, with components be separated by the token "[SEP]". KB triples are all instantiated with corresponding words in the knowledge base, where e_h, r, and e_t are separated by blanks. We use the top 3 triples in $\mathcal{K}(e)$ and feed them into the classifier, where the triples are sorted by their IDs. Such a choice is a trade-off between knowledge graph coverage and memory consumption in practice. If the number of relevant triples retrieved from the knowledge base is less than 3, we use the candidate entity itself to fill in the empty triples.

3.4 Type and Relation Prediction

This module mainly performs two subtasks: the unified recognition of entity types and relations, and the KB-guided prediction of correct entity types and relations stacked over the first subtask, as shown in the Type & Relation Prediction module in Figure 1.

Let $\mathcal{G} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ denote the knowledge base, where $\mathcal{E}$ is the entity set and $\mathcal{R}$ is the relation set. Each entity $e \in \mathcal{E}$ has an entity type $\tau \in \mathcal{T}$ (entity type set).

We model the type and relation recognition subtask as a multi-label classification task and use a classifier to predict the probability of an output sequence from a given input sequence.

To obtain neural representations of both entity types and entity relations for the recognition subtask, we use a pre-trained language model BERT (Devlin et al., 2019). The input fed into BERT is formed in a way similar to the entity disambiguation module. The difference is that we replace the entity with its entity type. Formally, the neural representation $\boldsymbol{e}^{\tau}$ of an entity type is computed as follows:

$$\boldsymbol{e}^{\tau} = \text{BERT}_{[\text{CLS}]}([\text{CLS}]s(\tau)[\text{SEP}]\mathcal{K}(\tau)[\text{SEP}]),$$

where [CLS] indicates that we use the representation of the prepended artificial [CLS] token as the representation of the entity type τ, $s(\tau)$ and $\mathcal{K}(\tau)$ represent the surface form and triples of τ, respectively. Similarly, the neural representation e^r of a relation is formulated as:

$$\boldsymbol{e}^r = \text{BERT}_{[\text{CLS}]}([\text{CLS}]s(r)[\text{SEP}]\mathcal{K}(r)[\text{SEP}]).$$

Kacupaj et al. (2021) find that modeling the correlations between entity types and relations is crucial for semantic parsing. In our KaFSP, we use a single classifier to predict both entity types and relations, instead of using two separate classifiers that share no common information (Shen et al., 2019; Kacupaj et al., 2021). Hence, the prediction space of our classifier is $\mathcal{T} \cup \mathcal{R}$, and the correlations between types and relations are naturally captured in the same single classifier. We use a sigmoid function to output probabilities as follows:

$$P(\boldsymbol{y}^{\text{MLC}}|\boldsymbol{x}) = \text{Sigmoid}(\boldsymbol{h}_{\text{ctx}}^{\text{enc}} \times \boldsymbol{W}^{\text{MLC}}(\boldsymbol{e}^{\tau r})^{\top}), \tag{7}$$

where $\boldsymbol{W}^{\text{MLC}} \in \mathbb{R}^{|\mathcal{T} \cup \mathcal{R}| \times d}$ is a linear projection matrix, and $\boldsymbol{e}^{\tau r}$ are the concatenation of the embeddings of $\tau \in \mathcal{T}$ and $r \in \mathcal{R}$.

The KB-guided prediction of entity types and relations is actually to make final decisions on them with relevant information from the knowledge base. Since KB contains a lot of triples irrelevant to the current utterance u, in order to make the knowledge graph embedding provide the information related to u, we use the output probabilities from the proposed multi-label classifier to pinpoint relevant information from the knowledge base encoded by GAT. Particularly, we calculate the Hadamard product of $P(\boldsymbol{y}^{\text{MLC}}|\boldsymbol{x})$ and $\boldsymbol{h}^{\text{GAT}}$:

$$\boldsymbol{h}^{\text{MLC}} = \boldsymbol{W}^{\text{TRP}}(\boldsymbol{h}^{\text{GAT}} \odot P(\boldsymbol{y}^{\text{MLC}}|\boldsymbol{x})), \tag{8}$$

where $\boldsymbol{W}^{\text{TRP}} \in \mathbb{R}^{2d \times d}$ is a linear projection matrix.

Given the hidden states of the decoder $\boldsymbol{z}^{\text{dec}}$ and last hidden state of the encoder $\boldsymbol{h}_{\text{ctx}}^{\text{enc}}$, we use a feedforward network to predict the sequence of types and relations:

$$P(\boldsymbol{y}^{\text{TRP}}|\boldsymbol{x}) = \prod_t \text{softmax}((\boldsymbol{h}^{\text{MLC}})^{\top}\text{FFN}(\boldsymbol{h}_{\text{ctx}}^{\text{enc}}; \boldsymbol{z}_t^{\text{dec}})), \tag{9}$$

where $\text{FFN}(\boldsymbol{h}_{\text{ctx}}^{\text{enc}}; \boldsymbol{z}_t^{\text{dec}})$ is the projection of the concatenation of the context representation and the hidden state of the decoder at time step t.

4 Learning and Inference

4.1 KaFSP Training

Before training KaFSP, we use weak supervisions (only the final answers) to obtain golden standard logical forms of questions in the training set through BFS, following Guo et al. (2018).

In KaFSP, we have 6 subtasks: the encoder-decoder subtask (DEC), the entity recognition subtask (ER), the filtering and permutation subtask from LASAGNE (FP), the multi-label classification subtask (MLC) described in Section 3.4, the type and relation prediction subtask (TRP) and the entity disambiguation subtask (ED). We a mixed training strategy to train these subtasks. The first 5 subtasks are jointly trained in a multi-task learning way while the last subtask is separately trained. Reasons for this strategy are twofold: 1) Entity disambiguation is a relatively independent subtask compared with other subtasks. 2) We fine-tune a huge pre-trained language model XLNet (Yang et al., 2019) on this subtask. Direct incorporation of the fine-tuning procedure into multi-task learning may make it difficult for the entire model to converge.

The joint loss $\mathcal{J}$ for the multi-task learning training is formulated as:

$$\mathcal{J} = \sum_{m \in \mathcal{M}} \gamma^m \mathcal{L}^m, \tag{10}$$

where $\mathcal{M} = \{\text{DEC}, \text{ER}, \text{FP}, \text{MLC}, \text{TRP}\}$ is the set of subtasks and γs are the weights of these subtasks, which are learned during training. In learning these weights, we take into account the difference in magnitude among the 5 losses according to the log standard deviation (Kendall et al., 2018). $\mathcal{L}^{\text{DEC}}$, $\mathcal{L}^{\text{ER}}$, $\mathcal{L}^{\text{FP}}$ and $\mathcal{L}^{\text{TRP}}$ are the negative log-likelihood losses of 4 subtasks, which are defined as follows:

$$\begin{aligned}
\mathcal{L}^{\text{DEC}} &= -\sum_{k=1}^{m} \log P(\boldsymbol{y}_k^{\text{DEC}}|\boldsymbol{x}), \\
\mathcal{L}^{\text{ER}} &= -\sum_{j=1}^{n} \log P(\boldsymbol{y}_j^{\text{ER}}|\boldsymbol{x}), \\
\mathcal{L}^{\text{FP}} &= -\sum_{j=1}^{n} \log P(\boldsymbol{y}_j^{\text{FP}}|\boldsymbol{x}), \\
\mathcal{L}^{\text{TRP}} &= -\sum_{k=1, \boldsymbol{y}_k^{\text{dec}} \in \mathbb{P}}^{m} \log P(\boldsymbol{y}_k^{\text{TRP}}|\boldsymbol{x}),
\end{aligned} \tag{11}$$

where n and m are the length of the input utterance $\boldsymbol{x}$ and the golden standard logical form, respectively. $\mathbb{P}$ is the set of placeholders for relations and types. $\boldsymbol{y}^*$ are ground-truth labels for corresponding subtasks.

The loss for the multi-label classification $\mathcal{L}^{\mathrm{MLC}}$ is a binary cross-entropy loss, defined as:

$$\begin{aligned}\mathcal{L}^{\mathrm{MLC}} = -\frac{1}{l}\sum_{i=1}^{l} & \boldsymbol{y}_i^{\mathrm{MLC}} \log(P(\boldsymbol{y}_i^{\mathrm{MLC}}|\boldsymbol{x})) \\ & + \bar{\boldsymbol{y}_i}^{\mathrm{MLC}} \log(P(\bar{\boldsymbol{y}_i}^{\mathrm{MLC}}|\boldsymbol{x})),\end{aligned} \tag{12}$$

where l is the size of $\mathcal{T} \cup \mathcal{R}$, $\bar{\boldsymbol{y}}^{\mathrm{MLC}} = 1 - \boldsymbol{y}^{\mathrm{MLC}}$, $\boldsymbol{y}^{\mathrm{MLC}}$ is defined in Eq. (7).

The entity disambiguation is trained separately, and its loss function is defined as:

$$\begin{aligned}\mathcal{L}^{\mathrm{ED}} = \sum_{e \in \mathcal{E}_{\boldsymbol{x}}} & \boldsymbol{y}_i^{\mathrm{ED}} \log(P(\boldsymbol{y}_i^{\mathrm{ED}}|\boldsymbol{x})) \\ & + \bar{\boldsymbol{y}_i}^{\mathrm{ED}} \log(P(\bar{\boldsymbol{y}_i}^{\mathrm{ED}}|\boldsymbol{x})),\end{aligned} \tag{13}$$

where $\mathcal{E}_{\boldsymbol{x}}$ is the set of entities that appear in $\boldsymbol{x}$ and $\boldsymbol{y}^{\mathrm{ED}}$ is defined in Eq. (6).

To train this subtask, we retrieve all entities that are present in the current input from the knowledge base. Note that we only construct 500,000 and 40,000 samples respectively for training and validation of the entity disambiguation module.

4.2 Grammar-Guided Inference

The grammar defined in Table 1 is used to guide the decoding step. The decoder generates a sequence mixed with actions and placeholders. Placeholders are instantiated with specific entities, types, relations, and numbers. The decoding process for a logical form terminates when no nonterminals remain. After decoding, we use a shift-reduce method to check the logical form sequence and delete or correct wrong placeholders.

Once the BIO tags and entity types are identified, entity spans can be located from the input utterance. We search from the inverted index constructed for the knowledge base for each predicted entity span to obtain an entity candidate list. After filtering the retrieved entity candidate list according to the corresponding entity type, if there are still multiple candidate entities, the entity disambiguation module is activated to calculate the conditional probability of each candidate entity. The candidate entity with the highest probability is selected.

Finally, we use the relation and type prediction results and disambiguated entities to instantiate the placeholders to get final logical forms.

5 Experiments

We carried out experiments and analyses to validate the effectiveness of the proposed KaFSP.

5.1 Experimental Settings

Dataset We evaluated the proposed model on the CSQA dataset (Saha et al., 2018), a standard dataset for complex sequential question answering. The dataset is composed of 200K dialogues with 1.6M turns, and over 12.8M entities from Wikidata, where 153K, 16K, and 28K dialogues are used for training, verification, and test, respectively. The questions cover a wide range of linguistic phenomena, such as co-reference, ellipsis, and reasoning.

Evaluation Metrics We used the same evaluation metrics as Saha et al. (2018). When answers are composed of one or more entities, F1 score is used as the evaluation metric. When answers are a Boolean value or number, accuracy is used as the metric. Following previous works (Guo et al., 2018; Shen et al., 2019; Kacupaj et al., 2021), we also calculated overall scores for all types of questions under each evaluation metric.

Baselines We compared KaFSP against 5 state-of-the-art baselines on the CSQA. The first baseline is HRED+KVM (Saha et al., 2018), which combines the HRED model with the key-value memory network. The other four baselines are D2A (Guo et al., 2018), MaSP (Shen et al., 2019), KISP (Thirukovalluru et al., 2021), LASAGNE (Kacupaj et al., 2021), which achieve state-of-the-art results on different types of questions on the CSQA dataset. More details for model settings can be found in Appendix A.

5.2 Results

Table 2 shows experiment results on the CSQA dataset. Our model outperforms LASAGNE on all types of questions and achieves new SOTA results in 8 out of 10 question types. Additionally, our model outperforms all previous baselines in terms of "overall" results.

For question types that involve one or more entities, namely *Logical Reasoning (All)*, *Simple Question (Direct)*, and *Verification (Boolean)*, the improvements over LASAGNE on these question types are 3.14%, 2.78%, and 1.29% respectively. This is mainly because we have added a knowledge-aware entity disambiguation module to improve the accuracy of entity linking.

Method # Params	HRED-KVM -	D2A -	MaSP 15M	KISP 160M	LASAGNE 14.7M	KaFSP 133.7M	Δ
Question Type	F1 score						
Overall	9.39%	66.70%	79.26%	83.01%	82.91%	**85.33%**	+2.42%
Clarification	16.35%	35.53%	80.79%	76.33%	69.46%	**81.37%**	+11.91%
Comparative Reasoning (All)	2.96%	45.85%	68.90%	67.83%	69.77%	**86.00%**	+16.23%
Logical Reasoning (All)	8.33%	67.31%	69.04%	87.14%	89.83%	**92.97%**	+3.14%
Quantitative Reasoning (All)	0.96%	56.41%	73.75%	77.52%	86.67%	**93.74%**	+7.07%
Simple Question (Co-referenced)	7.26%	57.69%	76.47%	**79.66%**	79.06%	79.61%	+0.55%
Simple Question (Direct)	13.64%	78.42%	85.18%	87.68%	87.95%	**90.73%**	+2.78%
Simple Question (Ellipsis)	9.95%	81.14%	83.73%	**86.06%**	80.09%	81.75%	+1.66%
Question Type	Accuracy						
Overall	14.95%	37.33%	45.56%	46.22%	64.34%	**69.32%**	+4.98%
Verification (Boolean)	21.04%	45.05%	60.63%	72.72%	78.86%	**80.15%**	+1.29%
Quantitative Reasoning (Count)	12.13%	40.94%	43.39%	50.92%	55.18%	**61.23%**	+6.05%
Comparative Reasoning (Count)	8.67%	17.78%	22.26%	27.32%	53.34%	**72.79%**	+19.45%

Table 2: Comparison results against 5 baselines on CSQA. The last column shows the absolute improvements of our model over LASAGNE.

For question types *Clarification*, *Comparative Reasoning (All)*, and *Comparative Reasoning (Count)*, they usually involve multiple entity types and relations. KaFSP achieves huge improvements of 11.91%, 16.23%, and 19.45% on these question types over LASAGNE. This is mainly due to fuzzy comparison rules in the new grammar system and the proposed knowledge-aware type and relation prediction module. The module benefits from the multi-label classification with a single classifier that not only helps to capture correlations between entity types and relations but also pinpoints and incorporates only relevant information from the knowledge base into relation and type prediction, which makes the predictions of types and relations more accurate.

Our model does not outperform previous SOTA results on only 2 question types, i.e., *Simple Question (Co-referenced)* and *Simple Question (Ellipsis)*. Although KaFSP is lower than KISP on these two question types, it is 0.55% and 1.66% higher than LASAGNE. We conjecture that the reasons for being not superior to KISP on these question types are twofold. First, spurious logical forms may have a negative impact on the decoder when it is trained on data indeed with false logical forms. Second, in conversational QA, not only entities but also entity relations can be omitted in questions. For example, "*How many people acted as an influence on Thomas Aquinas? And also tell me about Walt Whitman?*". In KaFSP, we replace the real ID of an omitted entity with "previous-entity". However, this strategy is not used for omitted relations when producing logic forms, which may have negative impacts on the two question types mentioned above.

Methods	KaFSP	w/o Fuzzy	w/o ED	w/o MLC
Question Type	F1 score			
Clarification	81.37%	69.96%	79.44%	79.17%
Comparative	86.00%	70.55%	85.88%	85.65%
Logical	92.97%	-	90.03%	89.60%
Quantitative	93.74%	86.64%	-	93.32%
Simple(Coref)	79.61%	-	77.94%	77.28%
Simple(Direct)	90.73%	-	88.13%	88.19%
Simple(Ellipsis)	81.75%	-	80.34%	79.05%
Question Type	Accuracy			
Verification	80.15%	-	78.15%	79.02%
Quantitative	61.23%	57.74%	59.36%	59.46%
Comparative	72.79%	54.55%	72.39%	71.93%

Table 3: Ablation Study. "-" means the result is the same as KaFSP.

Furthermore, although KaFSP increases the number of parameters, most added parameters are from the pretrained XLNet (base) model included for entity disambiguation. This does not have a big impact on the inference speed of KaFSP compared to LASAGNE.

5.3 Ablation Study

Table 3 summarizes experiment results of ablation study on our major contributions: fuzzy grammar, the knowledge-aware entity disambiguation module, and the multi-label classification framework. We observe that all three key components make substantial contributions to our proposed model.

For the ablation study on the entity disambiguation module, we compared KaFSP against "w/o ED" that directly selects the first entity from the ordered candidate list retrieved from the knowledge

Question Type	Disamb.	Total	Perc.
Total	38, 023	272, 060	13.98%
Clarification	1, 403	12, 121	11.57%
Comparative	1, 540	15, 284	10.08%
Logical	4, 982	22, 466	22.18%
Quantitative	0	9, 253	0%
Simple (Coref)	5, 061	54, 854	9.23%
Simple (Direct)	11, 570	81, 994	14.11%
Simple (Ellipsis)	1, 466	10, 045	14.59%
Verification	5, 961	26, 728	22.30%
Quantitative (Cnt)	4, 495	24, 281	18.51%
Comparative (Cnt)	1, 545	15, 034	10.28%

Table 4: The number of entities in different types of questions, the number and percentage of entities disambiguated by the proposed entity disambiguation module.

base as the disambiguated entity. When ED module isn't used, candidate entities are sorted lexicographically by their IDs. This was done to be consistent with previous approaches in our baselines. We find that for all types of questions, the application of the proposed knowledge-aware ED improves the results to various degrees. This is because entity ambiguity is present in a wide range of questions. For *Simple Question (Direct)* questions, our further analysis shows that 14.11% of entities are updated by our knowledge-aware ED, which leads to an improvement of 2.60%. Both natural language questions and the knowledge base contain information that can be used to disambiguate entities. The proposed knowledge-aware ED incorporates both types of information for disambiguation.

Table 4 shows the total number of entities, the total number of disambiguated entities, and their proportions in the logical forms of different types of questions. It can be seen that overall, the disambiguated entities account for 13.98%. For *Logical Reasoning* and *Verification* questions where the proportion of disambiguated entities is relatively high, correspondingly, the improvements achieved by adding the entity disambiguation module is high. This further validates the effectiveness of the proposed entity disambiguation module.

Similarly, our ablation study validates the effectiveness of both the fuzzy grammar and the knowledge-aware multi-label classification (case study on the multi-label classification can be found in Appendix B).

5.4 Error Analysis

For error analysis, we randomly sampled 100 incorrect predictions and summarized the following two types of typical errors:

Entity Ambiguity (54%) Although our entity disambiguation model can achieve a prediction accuracy of 95.16%, ambiguous entities still exist in some questions. Take the question *"What lead to the death of Jerry Stephenson?"* as an example. Both entity Q6184489 and Q100927364 are found in the knowledge base, which matches the surface form "*Jerry Stephenson*". However, it is difficult to determine whether the real entity in the question is Q100927364 (college basketball player (1971–1971) Austin Peay) or Q6184489 (American baseball player) with only information of three triples and insufficient context.

Spurious Logical Forms (6%) Similar to previous works (Shen et al., 2019; Kacupaj et al., 2021), we find that our model can infer correct answers even with wrong "ground-truth" logical forms generated with the algorithm taken from previous work (Guo et al., 2018). This will affect the overall performance of the model. Such a phenomenon is especially common in complex reasoning questions.

6 Conclusion

In this paper, we have presented a knowledge-aware fuzzy semantic parsing framework KaFSP for conversational question answering over a large-scale knowledge base. KaFSP defines fuzzy comparison actions in grammar based on the fuzzy set theory to cover approximately comparative reasoning. In addition to this, we propose two knowledge-aware components in KaFSP to incorporate information from the knowledge base for entity disambiguation and entity type & relation prediction. Experiment results demonstrate that KaFSP is substantially better than all previous state-of-the-art models, setting new SOTA results on 8 out of 10 question types on the CSQA dataset and achieving over 90% F1 or accuracy in 3 question types for the first time.

Acknowledgments

The present research was supported by Zhejiang Lab (No. 2022KH0AB01) and Huawei (No. TC20210528011). We would like to thank the anonymous reviewers for their insightful comments. We also want to thank MindSpore[1] for the partial suppoort of this work, which is a new deep learning computing framework.

[1] https://www.mindspore.cn/

References

Jacob Andreas, Andreas Vlachos, and Stephen Clark. 2013. Semantic parsing as machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 47–52.

Yoav Artzi and Luke Zettlemoyer. 2013. Weakly supervised learning of semantic parsers for mapping instructions to actions. *Transactions of the Association for Computational Linguistics*, 1:49–62.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-based question answering with knowledge graph. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2503–2514.

Nikita Bhutani, Xinyi Zheng, Kun Qian, Yunyao Li, and H Jagadish. 2020. Answering complex questions by combining information from curated and extracted knowledge bases. In *Proceedings of the First Workshop on Natural Language Interfaces*, pages 1–10.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Li Dong and Mirella Lapata. 2016. Language to logical form with neural attention. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33–43.

Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. In *56th Annual Meeting of the Association for Computational Linguistics*, pages 731–742. Association for Computational Linguistics.

Mohnish Dubey, Sourish Dasgupta, Ankit Sharma, Konrad Höffner, and Jens Lehmann. 2016. Asknow: A framework for natural language query formalization in sparql. In *European Semantic Web Conference*, pages 300–316. Springer.

Daya Guo, Duyu Tang, Nan Duan, Ming Zhou, and Jian Yin. 2018. Dialog-to-action: Conversational question answering over a large-scale knowledge base. In *Advances in Neural Information Processing Systems*, pages 2942–2951.

Robin Jia and Percy Liang. 2016. Data recombination for neural semantic parsing. *arXiv preprint arXiv:1606.03622*.

Endri Kacupaj, Joan Plepi, Kuldeep Singh, Harsh Thakkar, Jens Lehmann, and Maria Maleshkova. 2021. Conversational question answering over knowledge graphs with transformer and graph attention networks. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 850–862, Online. Association for Computational Linguistics.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491.

Tom Kwiatkowski, Luke Zettlemoyer, Sharon Goldwater, and Mark Steedman. 2011. Lexical generalization in ccg grammar induction for semantic parsing. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1512–1523.

Yunshi Lan and Jing Jiang. 2020. Query graph generation for answering multi-hop complex questions from knowledge bases. Association for Computational Linguistics.

Yunshi Lan and Jing Jiang. 2021. Modeling transitions of focal entities for conversational knowledge base question answering. 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)*, pages 3288–3297, Online. Association for Computational Linguistics.

Reginald Long, Panupong Pasupat, and Percy Liang. 2016. Simpler context-dependent logical forms via model projections. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1456–1465.

Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. 2016. Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *International semantic web conference*, pages 177–185. Springer.

Siva Reddy, Mirella Lapata, and Mark Steedman. 2014. Large-scale semantic parsing without question-answer pairs. *Transactions of the Association for Computational Linguistics*, 2:377–392.

Amrita Saha, Vardaan Pahuja, Mitesh M Khapra, Karthik Sankaranarayanan, and Sarath Chandar. 2018. Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Tao Shen, Xiubo Geng, Tao Qin, Daya Guo, Duyu Tang, Nan Duan, Guodong Long, and Daxin Jiang. 2019. Multi-task learning for conversational question answering over a large-scale knowledge base. In

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 2442–2451, Hong Kong, China. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Raghuveer Thirukovalluru, Mukund Sridhar, Dung Thai, Shruti Chanumolu, Nicholas Monath, Sankaranarayanan Ananthakrishnan, and Andrew McCallum. 2021. Knowledge informed semantic parsing for conversational question answering. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 231–240.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Yuk Wah Wong and Raymond Mooney. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 960–967.

Chunyang Xiao, Marc Dymetman, and Claire Gardent. 2016. Sequence-based structured prediction for semantic parsing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1341–1350.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic parsing for single-relation question answering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 643–648.

L.A. Zadeh. 1965. Fuzzy sets. *Information and Control*, 8(3):338–353.

Luke Zettlemoyer and Michael Collins. 2007. Online learning of relaxed ccg grammars for parsing to logical form. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 678–687.

Luke S Zettlemoyer and Michael Collins. 2009. Learning context-dependent mappings from sentences to logical form.

Kai Zhao and Liang Huang. 2015. Type-driven incremental semantic parsing with polymorphism. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1416–1421.

Hyper-parameters	value
epochs	16
batch size	50
dropout ratio	0.1
learning rate	0.001
warmup steps	4000
optumizer	Adam
β_1	0.9
β_2	0.999
ϵ	1e-09
model dimension	300
word embeddings	GloVe
non-linear activation	LeakyReLU
GAT input dimension	3072
GAT node dimension	300
GAT pre-trained embeddings	BERT
λ for the fuzzy set A	0.85
b for the membership function $\mu(x)$	1
c for the membership function $\mu(x)$	$\frac{\sqrt{51a}}{20}$

Table 5: Hyper-parameters for KaFSP.

A Hyperparamters and Module Configurations

Table 5 summarizes the hyperparameters used in the KaFSP framework. For the Transformer module, we used the standard configurations from Vaswani et al. (2017). Following the setting for Transformer base, we used residual dropout (Srivastava et al., 2014) in the summation of word embeddings and positional encodings in both the encoder and decoder with a rate of 0.1.

B Case Study on the Multi-Label Classification

We sample 12 questions and list the ground-truth labels (i.e., entity types and relations) corresponding to each question in Table 6. Through the proposed knowledge-aware multi-label classification module, we get the probabilities of different labels for each question, which are visualized in Figure 2.

We observe that the knowledge-aware multi-label classification module can effectively recognize entity types and relations in the questions. Take Question 10 and 11 as examples. Both questions contain type T2 and relation R1, while Question 11 has another relation R4. In Figure 2, we observe that this module can recognize these types and relations correctly. Applying the proposed multi-label classification module to the type and relation prediction module will hence effectively filter information irrelevant to the current question and help the model make better predictions.

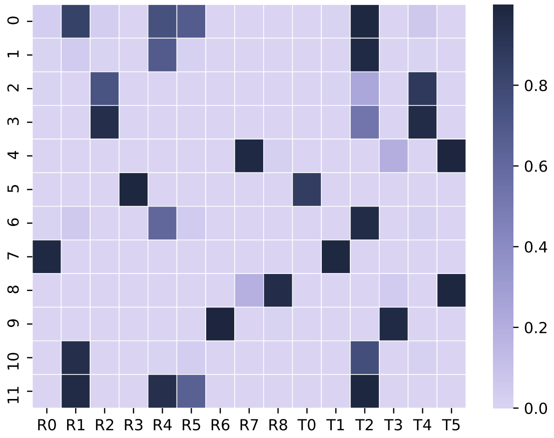

Figure 2: MLC probabilities of entity types and relations on sampled 12 questions. We represent questions with their IDs, and use T* and R* to represent Q* and P* which are the real IDs of the entity types and the relations in the Wikidata. Original questions with ground-truth entity types and relations are shown in Table 6.

We also find that some questions have a higher probability on a very small number of irrelevant labels. However, as the number of labels in KBs is large, having a high probability of only a few irrelevant labels will not greatly affect the results of the entire model.

ID	Question	Ground-truth Labels
0	Which administrative territories have diplomatic relationships with Greece and do Eliyahu Sasson belong to ?	P27 (R4), P530 (R5), Q15617994 (T2)
1	Which administrative territories are the countries of citizenship of Alfred Haighton or Thom Vink ?	P27 (R4), Q15617994 (T2)
2	Which administrative territories are the places of birth of Alfred Haighton or Dougie Bell ?	P19 (R2), Q56061 (T4)
3	Which administrative territories were Argo Arbeiter or Vera Hrochova born ?	P19 (R2), Q56061 (T4)
4	Which nucleic acid sequences are the species that have Pusl1 or SGCZ as their orthologous gene ?	P684 (R7), Q863908 (T5)
5	Which special fields are the medical specalities concerned with infectious disease or Borderline tuberculoid leprosy ?	P1995 (R3), Q1047113 (T0)
6	Which administrative territories are the countries of citizenship of Dinei or Jurgen Beck ?	P27 (R4), Q15617994 (T2)
7	Which occupations do Margaret Cavendish, Duchess of Newcastle-upon-Tyne and Jonathan Mayhew do for a living ?	P106 (R0), Q12737077 (T1)
8	Which nucleic acid sequences genes encode inorganic polyphosphate/ATP-NAD kinase CBU_1296 or formate dehydrogenase-specific chaperone VP1511 ?	P702 (R8), Q863908 (T5)
9	Which nutrients are involved in metabolic process and phosphorylation ?	P682 (R6), Q181394 (T3)
10	Which administrative territories are 1980 Winter Olympics or Ha Tien, Kien Giang present in ?	P17 (R1), Q15617994 (T2)
11	Which administrative territories are 1980 Winter Olympics a part of or are the native countriesof Dale M. Cochran ?	P17 (R1), P27 (R4), Q15617994 (T2)

Table 6: Sample questions, IDs, and their ground-truth labels (entity types and relations).