# Mixing Context Granularities for Improved Entity Linking on Question Answering Data across Entity Categories

**Daniil Sorokin** and **Iryna Gurevych**
Ubiquitous Knowledge Processing Lab (UKP) and Research Training Group AIPHES
Department of Computer Science, Technische Universität Darmstadt
www.ukp.tu-darmstadt.de
{sorokin|gurevych}@ukp.informatik.tu-darmstadt.de

## Abstract

The first stage of every knowledge base question answering approach is to link entities in the input question. We investigate entity linking in the context of a question answering task and present a jointly optimized neural architecture for entity mention detection and entity disambiguation that models the surrounding context on different levels of granularity.

We use the Wikidata knowledge base and available question answering datasets to create benchmarks for entity linking on question answering data. Our approach outperforms the previous state-of-the-art system on this data, resulting in an average 8% improvement of the final score. We further demonstrate that our model delivers a strong performance across different entity categories.

## 1 Introduction

Knowledge base question answering (QA) requires a precise modeling of the question semantics through the entities and relations available in the knowledge base (KB) in order to retrieve the correct answer. The first stage for every QA approach is entity linking (EL), that is the identification of entity mentions in the question and linking them to entities in KB. In Figure 1, two entity mentions are detected and linked to the knowledge base referents. This step is crucial for QA since the correct answer must be connected via some path over KB to the entities mentioned in the question.

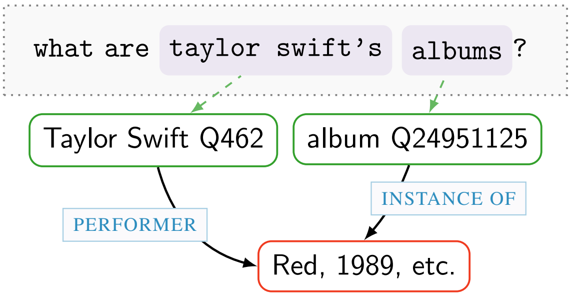

Figure 1: An example question from a QA dataset that shows the correct entity mentions and their relationship with the correct answer to the question, Qxxx stands for a knowledge base identifier

The state-of-the-art QA systems usually rely on off-the-shelf EL systems to extract entities from the question (Yih et al., 2015). Multiple EL systems are freely available and can be readily applied for question answering (e.g. DBPedia Spotlight[1], AIDA[2]). However, these systems have certain drawbacks in the QA setting: they are targeted at long well-formed documents, such as news texts, and are less suited for typically short and noisy question data. Other EL systems focus on noisy data (e.g. S-MART, Yang and Chang, 2015), but are not openly available and hence limited in their usage and application. Multiple error analyses of QA systems point to entity linking as a major external source of error (Berant and Liang, 2014; Reddy et al., 2014; Yih et al., 2015).

The QA datasets are normally collected from the web and contain very noisy and diverse data (Berant et al., 2013), which poses a number of challenges for EL. First, many common features used in EL systems, such as capitalization, are not meaningful on noisy data. Moreover, a question is a short text snippet that does not contain broader context that is helpful for entity disambiguation. The QA data also features many entities of various categories and differs in this respect from the Twitter datasets that are often used to evaluate EL systems.

[1] http://www.dbpedia-spotlight.org

[2] https://www.mpi-inf.mpg.de/yago-naga/aida/

*Proceedings of the 7th Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 65–75
New Orleans, June 5-6, 2018. ©2018 Association for Computational Linguistics

In this paper, we present an approach that tackles the challenges listed above: we perform entity mention detection and entity disambiguation jointly in a single neural model that makes the whole process end-to-end differentiable. This ensures that any token n-gram can be considered as a potential entity mention, which is important to be able to link entities of different categories, such as movie titles and organization names.

To overcome the noise in the data, we automatically learn features over a set of contexts of different granularity levels. Each level of granularity is handled by a separate component of the model. A token-level component extracts higher-level features from the whole question context, whereas a character-level component builds lower-level features for the candidate n-gram. Simultaneously, we extract features from the knowledge base context of the candidate entity: character-level features are extracted for the entity label and higher-level features are produced based on the entities surrounding the candidate entity in the knowledge graph. This information is aggregated and used to predict whether the n-gram is an entity mention and to what entity it should be linked.

**Contributions** The two main contributions of our work are:

(i) We construct two datasets to evaluate EL for QA and present a set of strong baselines: the existing EL systems that were used as a building block for QA before and a model that uses manual features from the previous work on noisy data.

(ii) We design and implement an entity linking system that models contexts of variable granularity to detect and disambiguate entity mentions. To the best of our knowledge, we are the first to present a unified end-to-end neural model for entity linking for noisy data that operates on different context levels and does not rely on manual features. Our architecture addresses the challenges of entity linking on question answering data and outperforms state-of-the-art EL systems.

**Code and Datasets** Our system can be applied on any QA dataset. The complete code as well as the scripts that produce the evaluation data can be found here: `https://github.com/UKPLab/starsem2018-entity-linking`.

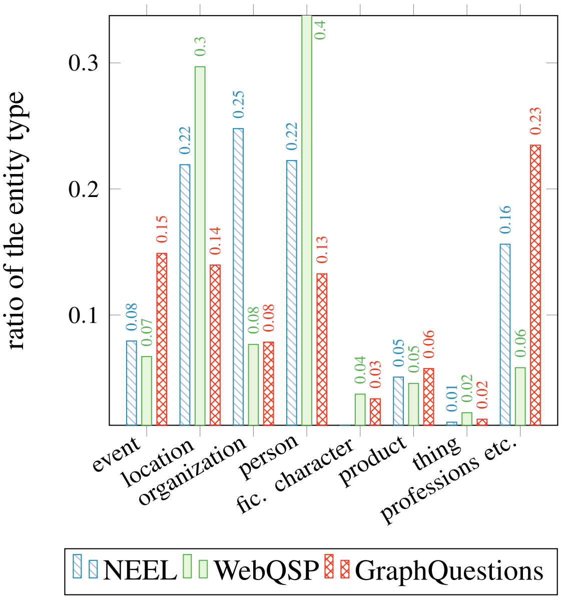

Figure 2: Distribution of entity categories in the NEEL 2014, WebQSP and GraphQuestions datasets

## 2 Motivation and Related Work

Several benchmarks exist for EL on Wikipedia texts and news articles, such as ACE (Bentivogli et al., 2010) and CoNLL-YAGO (Hoffart et al., 2011). These datasets contain multi-sentence documents and largely cover three types of entities: Location, Person and Organization. These types are commonly recognized by named entity recognition systems, such as Stanford NER Tool (Manning et al., 2014). Therefore in this scenario, an EL system can solely focus on entity disambiguation.

In the recent years, EL on Twitter data has emerged as a branch of entity linking research. In particular, EL on tweets was the central task of the NEEL shared task from 2014 to 2016 (Rizzo et al., 2017). Tweets share some of the challenges with QA data: in both cases the input data is short and noisy. On the other hand, it significantly differs with respect to the entity types covered. The data for the NEEL shared task was annotated with 7 broad entity categories, that besides Location, Organization and Person include Fictional Characters, Events, Products (such as electronic devices or works of art) and Things (abstract objects). Figure 2 shows the distribution of entity categories in the training set from the NEEL 2014 competition. One can see on the diagram that the distribution is mainly skewed towards 3 categories: Location, Person and Organization.

Figure 2 also shows the entity categories present in two QA datasets. The distribution over the categories is more diverse in this case. The WebQuestions dataset includes the Fictional Character and Thing categories which are almost absent from the NEEL dataset. A more even distribution can be observed in the GraphQuestion dataset that features many Events, Fictional Characters and Professions. This means that a successful system for EL on question data needs to be able to recognize and to link all categories of entities. Thus, we aim to show that comprehensive modeling of different context levels will result in a better generalization and performance across various entity categories.

**Existing Solutions** The early machine learning approaches to EL focused on long well-formed documents (Bunescu and Pasca, 2006; Cucerzan, 2007; Han and Sun, 2012; Francis-Landau et al., 2016). These systems usually rely on an off-the-shelf named entity recognizer to extract entity mentions in the input. As a consequence, such approaches can not handle entities of types other than those that are supplied by the named entity recognizer. Named entity recognizers are normally trained to detect mentions of Locations, Organizations and Person names, whereas in the context of QA, the system also needs to cover movie titles, songs, common nouns such as 'president' etc.

To mitigate this, Cucerzan (2012) has introduced the idea to perform mention detection and entity linking jointly using a linear combination of manually defined features. Luo et al. (2015) have adopted the same idea and suggested a probabilistic graphical model for the joint prediction. This is essential for linking entities in questions. For example in "*who does maggie grace play in taken?*", it is hard to distinguish between the usage of the word 'taken' and the title of a movie 'Taken' without consulting a knowledge base.

Sun et al. (2015) were among the first to use neural networks to embed the mention and the entity for a better prediction quality. Later, Francis-Landau et al. (2016) have employed convolutional neural networks to extract features from the document context and mixed them with manually defined features, though they did not integrate it with mention detection. Sil et al. (2018) continued the work in this direction recently and applied convolutional neural networks to cross-lingual EL.

The approaches that were developed for Twitter data present the most relevant work for EL on QA data. Guo et al. (2013b) have created a new dataset of around 1500 tweets and suggested a Structured SVM approach that handled mention detection and entity disambiguation together. Chang et al. (2014) describe the winning system of the NEEL 2014 competition on EL for short texts: The system adapts a joint approach similar to Guo et al. (2013b), but uses the MART gradient boosting algorithm instead of the SVM and extends the feature set. The current state-of-the-art system for EL on noisy data is S-MART (Yang and Chang, 2015) which extends the approach from Chang et al. (2014) to make structured predictions. The same group has subsequently applied S-MART to extract entities for a QA system (Yih et al., 2015).

Unfortunately, the described EL systems for short texts are not available as stand-alone tools. Consequently, the modern QA approaches mostly rely on off-the-shelf entity linkers that were designed for other domains. Reddy et al. (2016) have employed the Freebase online API that was since deprecated. A number of question answering systems have relied on DBPedia Spotlight to extract entities (Lopez et al., 2016; Chen et al., 2016). DBPedia Spotlight (Mendes et al., 2011) uses document similarity vectors, word embeddings and manually defined features such as entity frequency. We are addressing this problem in our work by presenting an architecture specifically targeted at EL for QA data.

**The Knowledge Base** Throughout the experiments, we use the Wikidata[3] open-domain KB (Vrandečić and Krötzsch, 2014). Among the previous work, the common choices of a KB include Wikipedia, DBPedia and Freebase. The entities in Wikidata directly correspond to the Wikipedia articles, which enables us to work with data that was previously annotated with DBPedia. Freebase was discontinued and is no longer up-to-date. However, most entities in Wikidata have been annotated with identifiers from other knowledge sources and databases, including Freebase, which establishes a link between the two KBs.

## 3 Entity Linking Architecture

The overall architecture of our entity linking system is depicted in Figure 3. From the input question $\mathbf{x}$ we extract all possible token n-grams $N$ up to a

[3] At the moment, Wikidata contains more than 40 million entities and 350 million relation instances: `https://www.wikidata.org/wiki/Special:Statistics`

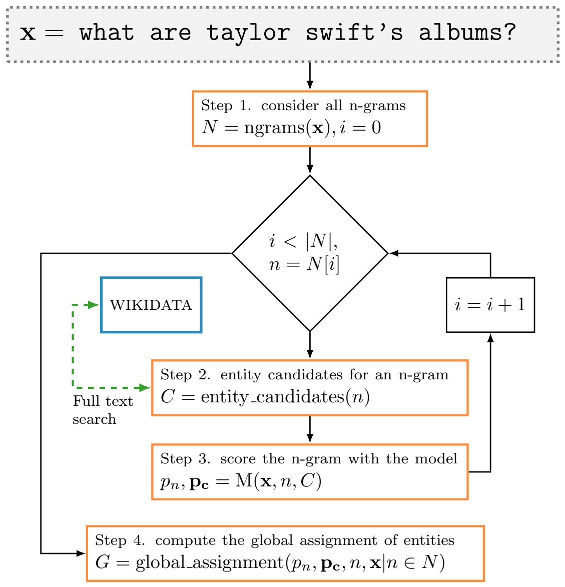
x = what are taylor swift's albums?
Step 1. consider all n-grams
$N = \text{ngrams}(\mathbf{x}), i = 0$
$i < |N|,$
$n = N[i]$
WIKIDATA
$i = i + 1$
Step 2. entity candidates for an n-gram
$C = \text{entity\_candidates}(n)$
Full text search
Step 3. score the n-gram with the model
$p_n, \mathbf{p_c} = \text{M}(\mathbf{x}, n, C)$
Step 4. compute the global assignment of entities
$G = \text{global\_assignment}(p_n, \mathbf{p_c}, n, \mathbf{x} | n \in N)$

Figure 3: Architecture of the entity linking system

certain length as entity mention candidates (Step 1). For each n-gram $n$, we look it up in the knowledge base using a full text search over entity labels (Step 2). That ensures that we find all entities that contain the given n-gram in the label. For example for a unigram 'obama', we retrieve 'Barack Obama', 'Michelle Obama' etc. This step produces a set of entity disambiguation candidates $C$ for the given n-gram $n$. We sort the retrieved candidates by length and cut off after the first 1000. That ensures that the top candidates in the list would be those that exactly match the target n-gram $n$.

In the next step, the list of n-grams $N$ and the corresponding list of entity disambiguation candidates are sent to the entity linking model (Step 3). The model jointly performs the detection of correct mentions and the disambiguation of entities.

### 3.1 Variable Context Granularity Network

The neural architecture (Variable Context Granularity, VCG) aggregates and mixes contexts of different granularities to perform a joint mention detection and entity disambiguation. Figure 4 shows the layout of the network and its main components.granularity level. The input to the model is a list of question tokens $\mathbf{x}$, a token n-gram $n$ and a list of candidate entities $C$. Then the model is a function $\text{M}(\mathbf{x}, n, C)$ that produces a mention detection score $p_n$ for each n-gram and a ranking score $p_c$ for each of the candidates $c \in C$: $p_n, \mathbf{p_c} = \text{M}(\mathbf{x}, n, C)$.

**Dilated Convolutions** To process sequential input, we use dilated convolutional networks (DCNN). Strubell et al. (2017) have recently shown that DCNNs are faster and as effective as recurrent models on the task of named entity recognition. We define two modules: **DCNN**$_w$ and **DCNN**$_c$ for processing token-level and character-level input respectively. Both modules consist of a series of convolutions applied with an increasing dilation, as described in Strubell et al. (2017). The output of the convolutions is averaged and transformed by a fully-connected layer.

**Context Components** The *token component* corresponds to sentence-level features normally defined for EL and encodes the list of question tokens $\mathbf{x}$ into a fixed size vector. It maps the tokens in $\mathbf{x}$ to $d_w$-dimensional pre-trained word embeddings, using a matrix $\mathbf{W} \in \mathbb{R}^{|V_w| \times d_w}$, where $|V_w|$ is the size of the vocabulary. We use 50-dimensional GloVe embeddings pre-trained on a 6 billion tokens corpus (Pennington et al., 2014). The word embeddings are concatenated with $d_p$-dimensional position embeddings $\mathbf{P_w} \in \mathbb{R}^{3 \times d_p}$ that are used to denote the tokens that are part of the target n-gram. The concatenated embeddings are processed by **DCNN**$_w$ to get a vector $\mathbf{o_s}$.

*The character component* processes the target token n-gram $n$ on the basis of individual characters. We add one token on the left and on the right to the target mention and map the string of characters to $d_z$-character embeddings, $\mathbf{Z} \in \mathbb{R}^{|V_z| \times d_z}$. We concatenate the character embeddings with $d_p$-dimensional position embeddings $\mathbf{P_z} \in \mathbb{R}^{|x| \times d_p}$ and process them with **DCNN**$_c$ to get a feature vector $\mathbf{o_n}$.

We use *the character component* with the same learned parameters to encode the label of a candidate entity from the KB as a vector $\mathbf{o_l}$. The parameter sharing between mention encoding and entity label encoding ensures that the representation of a mention is similar to the entity label.

The KB structure is the highest context level included in the model. *The knowledge base structure component* models the entities and relations that are connected to the candidate entity $c$. First, we map a list of relations $\mathbf{r}$ of the candidate entity to $d_r$-dimensional pre-trained relations embeddings, using a matrix $\mathbf{R} \in \mathbb{R}^{|V_r| \times d_r}$, where $|V_r|$ is the number of relation types in the KB. We transform the relations embeddings with a single fully-connected layer $f_r$ and then apply a max pooling operation to get a single relation vector $\mathbf{o_r}$ per entity. Similarly, we map a list of entities that are immediately connected to the candidate entity $\mathbf{e}$

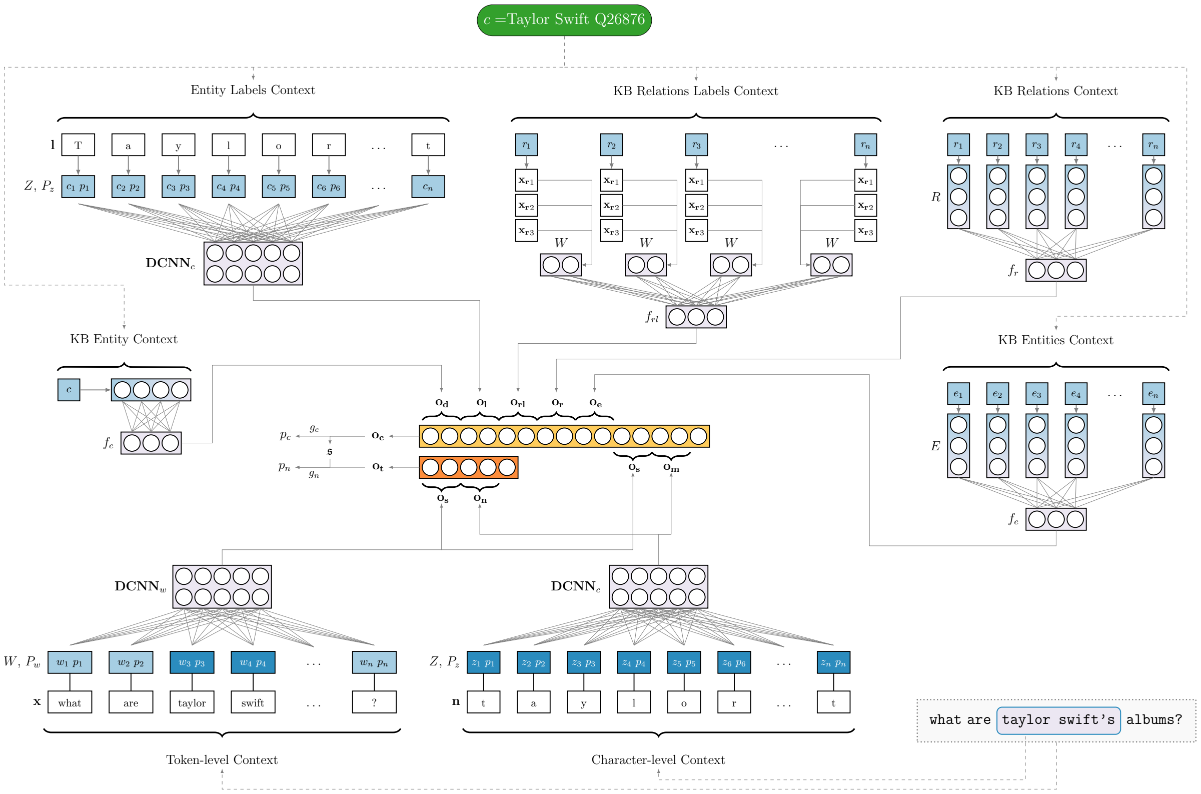
$c$ =Taylor Swift Q26876
Entity Labels Context
l
T a y l o r … t
$Z, P_z$
$c_1\ p_1$ $c_2\ p_2$ $c_3\ p_3$ $c_4\ p_4$ $c_5\ p_5$ $c_6\ p_6$ … $c_n$
DCNN$_c$
KB Relations Labels Context
$r_1$ $r_2$ $r_3$ … $r_n$
$\mathbf{x}_{r1}$ $\mathbf{x}_{r2}$ $\mathbf{x}_{r3}$
W
$f_{rl}$
KB Relations Context
$r_1$ $r_2$ $r_3$ $r_4$ … $r_n$
R
$f_r$
KB Entity Context
c
$f_e$
$\mathbf{o_d}$ $\mathbf{o_l}$ $\mathbf{o_{rl}}$ $\mathbf{o_r}$ $\mathbf{o_e}$
$p_c$ $g_c$ $\mathbf{o_c}$
$\mathfrak{s}$
$p_n$ $g_n$ $\mathbf{o_t}$
$\mathbf{o_s}$ $\mathbf{o_m}$
$\mathbf{o_s}$ $\mathbf{o_n}$
KB Entities Context
$e_1$ $e_2$ $e_3$ $e_4$ … $e_n$
E
$f_e$
DCNN$_w$
DCNN$_c$
$W, P_w$
$w_1\ p_1$ $w_2\ p_2$ $w_3\ p_3$ $w_4\ p_4$ … $w_n\ p_n$
x
what are taylor swift … ?
$Z, P_z$
$z_1\ p_1$ $z_2\ p_2$ $z_3\ p_3$ $z_4\ p_4$ $z_5\ p_5$ $z_6\ p_6$ … $z_n\ p_n$
n
t a y l o r … t
what are taylor swift's albums?
Token-level Context
Character-level Context

Figure 4: The architecture of the Variable Context Granularity Network for a *single* n-gram and an entity candidate. The output vectors $(\mathbf{o_c}, \mathbf{o_t})$ are aggregated over *all* n-grams for the global assignment

to $d_e$-dimensional pre-trained entity embeddings, using a matrix $\mathbf{E} \in \mathbb{R}^{|V_e| \times d_e}$, where $|V_e|$ is the number of entities in the KB. The entity embeddings are transformed by a fully-connected layer $f_e$ and then also pooled to produce the output $\mathbf{o_e}$. The embedding of the candidate entity itself is also transformed with $f_e$ and is stored as $\mathbf{o_d}$. To train the knowledge base embeddings, we use the TransE algorithm (Bordes et al., 2013).

Finally, *the knowledge base lexical component* takes the labels of the relations in $\mathbf{r}$ to compute lexical relation embeddings. For each $r \in \mathbf{r}$, we tokenize the label and map the tokens $\mathbf{x_r}$ to word embeddings, using the word embedding matrix $\mathbf{W}$. To get a single lexical embedding per relation, we apply max pooling and transform the output with a fully-connected layer $f_{rl}$. The lexical relation embeddings for the candidate entity are pooled into the vector $\mathbf{o_{rl}}$.

**Context Aggregation** The different levels of context are aggregated and are transformed by a sequence of fully-connected layers into a final vector $\mathbf{o_c}$ for the n-gram $n$ and the candidate entity $c$. The vectors for each candidate are aggregated into a matrix $O = [\mathbf{o_c} | c \in C]$. We apply element-wise max pooling on $O$ to get a single summary vector $\mathfrak{s}$ for all entity candidates for $n$.

To get the ranking score $p_c$ for each entity candidate $c$, we apply a single fully-connected layer $g_c$ on the concatenation of $\mathbf{o_c}$ and the summary vector $\mathfrak{s}$: $p_c = g_c(\mathbf{o_c} \| \mathfrak{s})$. For the mention detection score for the n-gram, we separately concatenate the vectors for the token context $\mathbf{o_s}$ and the character context $\mathbf{o_n}$ and transform them with an array of fully-connected layers into a vector $\mathbf{o_t}$. We concatenate $\mathbf{o_t}$ with the summary vector $\mathfrak{s}$ and apply another fully-connected layer to get the mention detection score $p_n = \sigma(g_n(\mathbf{o_t} \| \mathfrak{s}))$.

### 3.2 Global Entity Assignment

The first step in our system is extracting all possible overlapping n-grams from the input texts. We assume that each span in the input text can only refer to a single entity and therefore resolve overlaps by computing a global assignment using the model scores for each n-gram (Step 4 in Figure 3).

If the mention detection score $p_n$ is above the 0.5-threshold, the n-gram is predicted to be a correct entity mention and the ranking scores $\mathbf{p_c}$ are used to disambiguate it to a single entity candidate.

N-grams that have $p_n$ lower than the threshold are filtered out.

We follow Guo et al. (2013a) in computing the global assignment and hence, arrange all n-grams selected as mentions into non-overlapping combinations and use the individual scores $p_n$ to compute the probability of each combination. The combination with the highest probability is selected as the final set of entity mentions. We have observed in practice a similar effect as descirbed by Strubell et al. (2017), namely that DCNNs are able to capture dependencies between different entity mentions in the same context and do not tend to produce overlapping mentions.

### 3.3 Composite Loss Function

Our model jointly computes two scores for each n-gram: the mention detection score $p_n$ and the disambiguation score $p_c$. We optimize the parameters of the whole model jointly and use the loss function that combines penalties for the both scores for all n-grams in the input question:

$$\mathscr{L} \quad = \quad \sum_{n \in N} \sum_{c \in C_n} \mathscr{M}(t_n, p_n) \quad + \quad t_n \mathscr{D}(t_c, p_c),$$

where $t_n$ is the target for mention detection and is either 0 or 1, $t_c$ is the target for disambiguation and ranges from 0 to the number of candidates $|C|$.

For the mention detection loss $\mathscr{M}$, we include a weighting parameter $\alpha$ for the negative class as the majority of the instances in the data are negative:

$$\mathscr{M}(t_n, p_n) = -t_n \log p_n - \alpha(1 - t_n) \log(1 - p_n)$$

The disambiguation detection loss $\mathscr{D}$ is a maximum margin loss:

$$\mathscr{D}(t_c, p_c) = \frac{\sum_{i=0}^{|C|} \max(0, (m - p_c[t_c] + p_c[i]))}{|C|},$$

where $m$ is the margin value. We set $m = 0.5$, whereas the $\alpha$ weight is optimized with the other hyper-parameters.

### 3.4 Architecture Comparison

Our model architecture follows some of the ideas presented in Francis-Landau et al. (2016): they suggest computing a similarity score between an entity and the context for different context granularities. Francis-Landau et al. (2016) experiment on entity linking for Wikipedia and news articles and consider the word-level and document-level contexts for entity disambiguation. As described above, we also incorporate different context granularities with a number of key differences: (1) we operate on sentence level, word level and character level, thus including a more fine-grained range of contexts; (2) the knowledge base contexts that Francis-Landau et al. (2016) use are the Wikipedia title and the article texts — we, on the other hand, employ the structure of the knowledge base and encode relations and related entities; (3) Francis-Landau et al. (2016) separately compute similarities for each type of context, whereas we mix them in a single end-to-end architecture; (4) we do not rely on manually defined features in our model.

| | #Questions | #Entities |
|---|---|---|
| WebQSP Train | 3098 | 3794 |
| WebQSP Test | 1639 | 2002 |
| GraphQuestions Test | 2608 | 4680 |

Table 1: Dataset statistics

## 4 Datasets

We compile two new datasets for entity linking on questions that we derive from publicly available question answering data: WebQSP (Yih et al., 2016) and GraphQuestions (Su et al., 2016).

WebQSP contains questions that were originally collected for the WebQuestions dataset from web search logs (Berant et al., 2013). They were manually annotated with SPARQL queries that can be executed to retrieve the correct answer to each question. Additionally, the annotators have also selected the main entity in the question that is central to finding the answer. The annotations and the query use identifiers from the Freebase knowledge base.

We extract all entities that are mentioned in the question from the SPARQL query. For the main entity, we also store the correct span in the text, as annotated in the dataset. In order to be able to use Wikidata in our experiments, we translate the Freebase identifiers to Wikidata IDs.

The second dataset, GraphQuestions, was created by collecting manual paraphrases for automatically generated questions (Su et al., 2016). The dataset is meant to test the ability of the system to understand different wordings of the same question. In particular, the paraphrases include various references to the same entity, which creates a challenge for an entity linking system. The following

|  | P | R | F1 |
|---|---|---|---|
| Heuristic baseline | 0.286 | 0.621 | 0.392 |
| Simplified VCG | 0.804 | 0.654 | 0.721 |
| **VCG** | 0.823 | 0.646 | 0.724 |

Table 2: Evaluation results on the WEBQSP development dataset (all entities)

are three example questions from the dataset that contain a mention of the same entity:

(1) a. what is the rank of marvel's **iron man**?
b. **iron-man** has held what ranks?
c. **tony stark** has held what ranks?

GraphQuestions does not contain main entity annotations, but includes a SPARQL query structurally encoded in JSON format. The queries were constructed manually by identifying the entities in the question and selecting the relevant KB relations. We extract gold entities for each question from the SPARQL query and map them to Wikidata.

We split the WebQSP training set into train and development subsets to optimize the neural model. We use the GraphQuestions only in the evaluation phase to test the generalization power of our model. The sizes of the constructed datasets in terms of the number of questions and the number of entities are reported in Table 1. In both datasets, each question contains at least one correct entity mention.

# 5 Experiments

## 5.1 Evaluation Methodology

We use precision, recall and F1 scores to evaluate and compare the approaches. We follow Carmel et al. (2014) and Yang and Chang (2015) and define the scores on a per-entity basis. Since there are no mention boundaries for the gold entities, an extracted entity is considered correct if it is present in the set of the gold entities for the given question. We compute the metrics in the micro and macro setting. The macro values are computed per entity class and averaged afterwards.

For the WebQSP dataset, we additionally perform a separate evaluation using only the information on the main entity. The main entity has the information on the boundary offsets of the correct mentions and therefore for this type of evaluation, we enforce that the extracted mention has to overlap with the correct mention. QA systems need at least one entity per question to attempt to find the correct answer. Thus, evaluating using the main entity shows how the entity linking system fulfills this minimum requirement.

| emb. size |  |  |  |  | filter size |  |  |
|---|---|---|---|---|---|---|---|
| $d_w$ | $d_z$ | $d_e$ | $d_r$ | $d_p$ | **DCNN**$_w$ | **DCNN**$_c$ | $\alpha$ |
| 50 | 25 | 50 | 50 | 5 | 64 | 64 | 0.5 |

Table 3: Best configuration for the VCG model

## 5.2 Baselines

**Existing Systems** In our experiments, we compare to DBPedia Spotlight that was used in several QA systems and represents a strong baseline for entity linking[4]. In addition, we are able to compare to the state-of-the-art S-MART system, since their output on the WebQSP datasets was publicly released[5]. The S-MART system is not openly available, it was first trained on the NEEL 2014 Twitter dataset and later adapted to the QA data (Yih et al., 2015).

We also include a heuristics baseline that ranks candidate entities according to their frequency in Wikipedia. This baseline represents a reasonable lower bound for a Wikidata based approach.

**Simplified VCG** To test the effect of the end-to-end context encoders of the VCG network, we define a model that instead uses a set of features commonly suggested in the literature for EL on noisy data. In particular, we employ features that cover (1) frequency of the entity in Wikipedia, (2) edit distance between the label of the entity and the token n-gram, (3) number of entities and relations immediately connected to the entity in the KB, (4) word overlap between the input question and the labels of the connected entities and relations, (5) length of the n-gram. We also add an average of the word embeddings of the question tokens and, separately, an average of the embeddings of tokens of entities and relations connected to the entity candidate. We train the simplified VCG model by optimizing the same loss function in Section 3.3 on the same data.

## 5.3 Practical Considerations

The hyper-parameters of the model, such as the dimensionality of the layers and the size of embed-

[4] We use the online end-point: http://www.dbpedia-spotlight.org/api

[5] https://github.com/scottyih/STAGG

|  | Main entity |  |  | All entities |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
|  | P | R | F1 | P | R | F1 | mP | mR | mF1 |
| DBPedia Spotlight | 0.668 | 0.595 | 0.629 | 0.705 | 0.514 | 0.595 | 0.572 | 0.392 | 0.452 |
| S-MART | 0.634 | **0.899** | 0.744 | 0.666 | **0.772** | 0.715 | 0.607 | **0.610** | 0.551 |
| Heuristic baseline | 0.282 | 0.694 | 0.401 | 0.302 | 0.608 | 0.404 | 0.330 | 0.537 | 0.378 |
| Simplified VCG | **0.804** | 0.728 | 0.764 | **0.837** | 0.621 | 0.713 | 0.659 | 0.494 | 0.546 |
| **VCG** | 0.793 | 0.766 | **0.780** | 0.826 | 0.653 | **0.730** | **0.676** | 0.519 | **0.568** |

Table 4: Evaluation results on the WEBQSP test dataset, the `m` prefix stands for *macro*

|  | P | R | F1 |
|---|---|---|---|
| DBPedia Spotlight | 0.386 | 0.453 | 0.417 |
| **VCG** | **0.589** | 0.354 | **0.442** |

Table 5: Evaluation results on GRAPHQUESTIONS

dings, are optimized with random search on the development set. The model was particularly sensitive to tuning of the negative class weight $\alpha$ (see Section 3.3). Table 3 lists the main selected hyper-parameters for the VCG model[6] and we also report the results for each model's best configuration on the development set in Table 2.

### 5.4 Results

Table 4 lists results for the heuristics baseline, for the suggested Variable Context Granularity model (VCG) and for the simplified VCG baseline on the test set of WebQSP. The simplified VCG model outperforms DBPedia Spotlight and achieves a result very close to the S-MART model. Considering only the main entity, the simplified VCG model produces results better than both DBPedia Spotlight and S-MART. The VCG model delivers the best F-score across the all setups. We observe that our model achieves the most gains in precision compared to the baselines and the previous state-of-the-art for QA data.

VCG constantly outperforms the simplified VCG baseline that was trained by optimizing the same loss function but uses manually defined features. Thereby, we confirm the advantage of the mixing context granularities strategy that was suggested in this work. Most importantly, the VCG model achieves the best macro result which indicates that the model has a consistent performance on different entity classes.

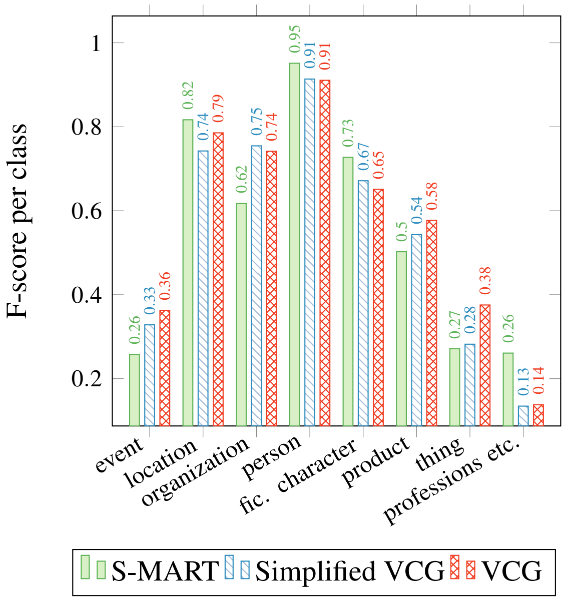

Figure 5: Performance accross entity classes on WEBQSP test dataset

We further evaluate the developed VCG architecture on the GraphQuestions dataset against the DBPedia Spotlight. We use this dataset to evaluate VCG in an out-of-domain setting: neither our system nor DBPedia Spotlight were trained on it. The results for each model are presented in Table 5. We can see that GraphQuestions provides a much more difficult benchmark for EL. The VCG model shows the overall F-score result that is better than the DBPedia Spotlight baseline by a wide margin. It is notable that again our model achieves higher precision values as compared to other approaches and manages to keep a satisfactory level of recall.

**Analysis** In order to better understand the performance difference between the approaches and the gains of the VCG model, we analyze the results per entity class (see Figure 5). We see that the S-MART system is slightly better in the disambiguation of Locations, Person names and a similar

[6] The complete list of hyper-parameters and model characteristics can be found in the accompanying code repository.

|  | Main entity |  |  | All entities |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
|  | P | R | F1 | P | R | F1 | mP | mR | mF1 |
| **VCG** | 0.793 | **0.766** | **0.780** | **0.826** | **0.653** | **0.730** | **0.676** | **0.519** | **0.568** |
| w/o token context | 0.782 | 0.728 | 0.754 | 0.812 | 0.618 | 0.702 | 0.664 | 0.474 | 0.530 |
| w/o character context | **0.802** | 0.684 | 0.738 | 0.820 | 0.573 | 0.675 | 0.667 | 0.404 | 0.471 |
| w/o KB structure context | 0.702 | 0.679 | 0.690 | 0.728 | 0.576 | 0.643 | 0.549 | 0.427 | 0.461 |
| w/o KB lexical context | 0.783 | 0.732 | 0.756 | 0.807 | 0.617 | 0.699 | 0.643 | 0.454 | 0.508 |

Table 6: Ablation experiments for the VCG model on WebQSP

category of Fictional Character names, while it has a considerable advantage in processing of Professions and Common Nouns. Our approach has an edge in such entity classes as Organization, Things and Products. The latter category includes movies, book titles and songs, which are particularly hard to identify and disambiguate since any sequence of words can be a title. VCG is also considerably better in recognizing Events. We conclude that the future development of the VCG architecture should focus on the improved identification and disambiguation of professions and common nouns.

To analyze the effect that mixing various context granularities has on the model performance, we include ablation experiment results for the VCG model (see Table 6). We report the same scores as in the main evaluation but without individual model components that were described in Section 3.

We can see that the removal of the KB structure information encoded in entity and relation embeddings results in the biggest performance drop of almost 10 percentage points. The character-level information also proves to be highly important for the final state-of-the-art performance. These aspects of the model (the comprehensive representation of the KB structure and the character-level information) are two of the main differences of our approach to the previous work. Finally, we see that excluding the token-level input and the lexical information about the related KB relations also decrease the results, albeit less dramatically.

## 6 Conclusions

We have described the task of entity linking on QA data and its challenges. The suggested new approach for this task is a unifying network that models contexts of variable granularity to extract features for mention detection and entity disambiguation. This system achieves state-of-the-art results on two datasets and outperforms the previous best system used for EL on QA data. The results further verify that modeling different types of context helps to achieve a better performance across various entity classes (macro f-score).

Most recently, Peng et al. (2017) and Yu et al. (2017) have attempted to incorporate entity linking into a QA model. This offers an exciting future direction for the Variable Context Granularity model.

## Acknowledgments

This work has been supported by the German Research Foundation as part of the Research Training Group Adaptive Preparation of Information from Heterogeneous Sources (AIPHES) under grant No. GRK 1994/1, and via the QA-EduInf project (grant GU 798/18-1 and grant RI 803/12-1).

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

## References

Luisa Bentivogli, Pamela Forner, Claudio Giuliano, Alessandro Marchetti, Emanuele Pianta, and Kateryna Tymoshenko. 2010. Extending English ACE 2005 corpus annotation with ground-truth links to Wikipedia. In *Proceedings of the 2nd Workshop on Collaboratively Constructed Semantic Resources at the 23rd International Conference on Computational Linguistics (Coling)*. Beijing, China, pages 19–26. http://www.aclweb.org/anthology/W10-3503.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Seattle, WA, USA, pages 1533–1544. http://www.aclweb.org/anthology/D13-1160.

Jonathan Berant and Percy Liang. 2014. Semantic Parsing via Paraphrasing. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*. Baltimore, MD, USA, pages 1415–1425. https://doi.org/10.3115/v1/P14-1133.

Antoine Bordes, Nicolas Usunier, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-Relational Data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS)*. Curran Associates, Inc., Lake Tahoe, NV, USA, volume 26, pages 2787–2795.

Razvan Bunescu and Marius Pasca. 2006. Using Encyclopedic Knowledge for Named Entity Disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*. Trento, Italy, pages 9–16. http://aclweb.org/anthology/E06-1002.

David Carmel, Ming-Wei Chang, Evgeniy Gabrilovich, Bo-June (Paul) Hsu, and Kuansan Wang. 2014. ERD'14: Entity Recognition and Disambiguation Challenge. In *ACM SIGIR Forum*. volume 48, pages 63–77. https://doi.org/10.1145/2600428.2600734.

Ming-Wei Chang, Bo-June Hsu, Hao Ma, Ricky Loynd, and Kuansan Wang. 2014. E2E: An End-to-End Entity Linking System for Short and Noisy text. In *Proceedings of the the 4thWorkshop on Making Sense of Microposts co-located with the 23rd International World Wide Web Conference (WWW)*. Seoul, Korea, pages 62–63.

Long Chen, Joemon M. Jose, Haitao Yu, Fajie Yuan, and Dell Zhang. 2016. A Semantic Graph based Topic Model for Question Retrieval in Community Question Answering. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining (WSDM)*. San Francisco, CA, USA, pages 287–296. https://doi.org/10.1145/2835776.2835809.

Silviu Cucerzan. 2007. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic, pages 708–716. http://aclweb.org/anthology/D07-1074.

Silviu Cucerzan. 2012. The MSR System for Entity Linking at TAC 2012. In *Proceedings of the Text Analysis Conference (TAC)*. Gaithersburg, MD, USA, pages 14–15.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. Capturing Semantic Similarity for Entity Linking with Convolutional Neural Networks. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. San Diego, CA, USA, pages 1256–1261. https://doi.org/10.18653/v1/N16-1150.

Stephen Guo, Ming-Wei Chang, and Emre Kcman. 2013a. To Link or Not to Link? A Study on End-to-End Tweet Entity Linking. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*. Atlanta, GA, USA, pages 1020–1030. http://www.aclweb.org/anthology/N13-1122.

Yuhang Guo, Bing Qin, Ting Liu, and Sheng Li. 2013b. Microblog entity linking by leveraging extra posts. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 863–868. http://www.aclweb.org/anthology/D13-1085.

Xianpei Han and Le Sun. 2012. An entity-topic model for entity linking. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Jeju Island, Korea, pages 105–115. http://www.aclweb.org/anthology/D12-1010.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust Disambiguation of Named Entities in Text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Edinburgh, UK, pages 782–792. http://www.aclweb.org/anthology/D11-1072.

Vanessa Lopez, Pierpaolo Tommasi, Spyros Kotoulas, and Jiewen Wu. 2016. QuerioDALI: Question answering over dynamic and linked knowledge graphs. In *The Semantic Web - 15th International Semantic Web Conference (ISWC 2016)*. Springer International Publishing, Kobe, Japan, pages 363–382. https://doi.org/10.1007/978-3-319-46547-0_32.

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint Named Entity Recognition and Disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Lisbon, Portugal. https://doi.org/10.18653/v1/D15-1104.

Christopher D. Manning, John Bauer, Jenny Finkel, Steven J. Bethard, Mihai Surdeanu, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics (ACL): System Demonstrations*. Baltimore, MD, USA, pages 55–60. https://doi.org/10.3115/v1/P14-5010.

Pablo N. Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. 2011. DBpedia Spotlight: Shedding Light on the Web of Documents. In *Proceedings of the 7th International Conference on Semantic Systems (I-Semantics)*. Graz, Austria, volume 95, pages 1–8. https://doi.org/10.1145/2063518.2063519.

Haoruo Peng, Ming-Wei Chang, and Wen-Tau Yih. 2017. Maximum Margin Reward Networks for

Learning from Explicit and Implicit Supervision. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Copenhagen, Denmark, pages 2358–2368. https://doi.org/10.18653/v1/D17-1252.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. https://doi.org/10.3115/v1/D14-1162.

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. http://aclweb.org/anthology/Q14-1030.

Siva Reddy, Oscar Täckström, Michael Collins, Tom Kwiatkowski, Dipanjan Das, Mark Steedman, and Mirella Lapata. 2016. Transforming Dependency Structures to Logical Forms for Semantic Parsing. *Transactions of the Association for Computational Linguistics* 4:127–140. http://aclweb.org/anthology/Q16-1010.

Giuseppe Rizzo, Bianca Pereira, Andrea Varga, Marieke Van Erp, and Amparo Elizabeth Cano Basave. 2017. Lessons learnt from the Named Entity rEcognition and Linking (NEEL) Challenge Series. *Semantic Web* 8(5):667–700. https://doi.org/10.3233/SW-170276.

Avirup Sil, Gourab Kundu, Radu Florian, and Wael Hamza. 2018. Neural Cross-Lingual Entity Linking. In *Association for the Advancement of Artificial Intelligence (AAAI)*. New Orleans, LA, US.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and Accurate Entity Recognition with Iterated Dilated Convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Copenhagen, Denmark, pages 2670–2680. https://doi.org/10.18653/v1/D17-1283.

Yu Su, Huan Sun, Brian Sadler, Mudhakar Srivatsa, Izzeddin Gur, Zenghui Yan, and Xifeng Yan. 2016. On Generating Characteristic-rich Question Sets for QA Evaluation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Austin, Texas, pages 562–572. https://doi.org/10.18653/v1/D16-1054.

Yaming Sun, Lei Lin, Duyu Tang, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2015. Modeling mention, context and entity with neural networks for entity disambiguation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. Buenos Aires, Argentina, pages 1333–1339.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A Free Collaborative Knowledgebase. *Communications of the ACM* 57(10):78–85. https://doi.org/10.1021/ac60289a702.

Yi Yang and Ming-Wei Chang. 2015. S-MART: Novel Tree-based Structured Learning Algorithms Applied to Tweet Entity Linking. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*. Beijing, China, pages 504–513. https://doi.org/10.3115/v1/P15-1049.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*. Beijing, China, pages 1321–1331. https://doi.org/10.3115/v1/P15-1128.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The Value of Semantic Parse Labeling for Knowledge Base Question Answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*. Berlin, Germany, pages 201–206. https://doi.org/10.18653/v1/P16-2033.

Mo Yu, Wenpeng Yin, Kazi Saidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved Neural Relation Detection for Knowledge Base Question Answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*. Vancouver, Canada, pages 571–581. https://doi.org/10.18653/v1/P17-1053.