CFO: Conditional Focused Neural Question Answering with Large-scale Knowledge Bases

Zihang Dai[*]
Carnegie Mellon University
dzihang@andrew.cmu.edu

Lei Li[*]
Toutiao.com
lileicc@gmail.com

Wei Xu
Baidu Research
xuwei06@baidu.com

Abstract

How can we enable computers to automatically answer questions like "*Who created the character Harry Potter*"? Carefully built knowledge bases provide rich sources of facts. However, it remains a challenge to answer factoid questions raised in natural language due to numerous expressions of one question. In particular, we focus on the most common questions — ones that can be answered with a single fact in the knowledge base. We propose CFO, a **C**onditional **Fo**cused neural-network-based approach to answering factoid questions with knowledge bases. Our approach first zooms in a question to find more probable candidate subject mentions, and infers the final answers with a unified conditional probabilistic framework. Powered by deep recurrent neural networks and neural embeddings, our proposed CFO achieves an accuracy of 75.7% on a dataset of 108k questions – the largest public one to date. It outperforms the current state of the art by an absolute margin of 11.8%.

1 Introduction

Community-driven question answering (QA) websites such as Quora, Yahoo-Answers, and Answers.com are accumulating millions of users and hundreds of millions of questions. A large portion of the questions are about facts or trivia. It has been a long pursuit to enable machines to answer such questions automatically.

In recent years, several efforts have been made on utilizing open-domain knowledge bases to answer factoid questions. A knowledge base (KB) consists of structured representation of facts in the form of subject-relation-object triples. Lately, several large-scale general-purpose KBs have been constructed, including YAGO (Suchanek et al., 2007), Freebase (Bollacker et al., 2008), NELL (Carlson et al., 2010), and DBpedia (Lehmann et al., 2014). Typically, structured queries with predefined semantics (e.g. SPARQL) can be issued to retrieve specified facts from such KBs. Thus, answering factoid questions will be straightforward once they are converted into the corresponding structured form. However, due to complexity of language, converting natural language questions to structure forms remains an open challenge.

Among all sorts of questions, there is one category that only requires a single fact (triple) in KB as the supporting evidence. As a typical example, the question "*Who created the character Harry Potter*" can be answered with the single fact `(HarryPotter, CharacterCreatedBy, J.K.Rowling)`. In this work, we refer to such questions as *single-fact questions*. Previously, it has been observed that *single-fact questions* constitute the majority of factoid questions in community QA sites (Fader et al., 2013). Despite the simplicity, automatically answering such questions remains far from solved — the latest best result on a dataset of 108k single-fact questions is only 63.9% in terms of accuracy (Bordes et al., 2015).

To find the answer to a single-fact question, it suffices to identify the subject entity and relation (implicitly) mentioned by the question, and then forms a corresponding structured query. The problem can be formulated into a probabilistic form. Given a single-fact question q, finding the subject-relation pair $\hat{s}, \hat{r}$ from the KB $\mathcal{K}$ which maximizes the conditional probability $p(s, r|q)$, i.e.

$$\hat{s}, \hat{r} = \arg\max_{s,r \in \mathcal{K}} p(s, r|q) \quad (1)$$

[*] Part of the work was done while at Baidu.

Based on the formulation (1), the central problem is to estimate the conditional distribution $p(s, r|q)$. It is very challenging because of *a*) the vast amount of facts — a large-scale KB such as Freebase contains billions of triples, *b*) the huge variety of language — there are multiple aliases for an entity, and numerous ways to compose a question, *c*) the severe sparsity of supervision — most combinations of s, r, q are not expressed in training data. Faced with these challenges, existing methods have exploited to incorporate prior knowledge into semantic parsers, to design models and representations with better generalization property, to utilize large-margin ranking objective to estimate the model parameters, and to prune the search space during inference. Noticeably, models based on neural networks and distributed representations have largely contributed to the recent progress (see section 2).

In this paper, we propose CFO, a novel method to answer *single-fact questions* with large-scale knowledge bases. The contributions of this paper are,

- we employ a fully probabilistic treatment of the problem with a novel conditional parameterization using neural networks,
- we propose the focused pruning method to reduce the search space during inference, and
- we investigate two variations to improve the generalization of representations for millions of entities under highly sparse supervision.

In experiments, CFO achieves 75.7% in terms of top-1 accuracy on the largest dataset to date, outperforming the current best record by an absolute margin of 11.8%.

2 Related Work

The research of KB supported QA has evolved from earlier domain-specific QA (Zelle and Mooney, 1996; Tang and Mooney, 2001; Liang et al., 2013) to open-domain QA based on large-scale KBs. An important line of research has been trying to tackle the problem by semantic parsing, which directly parses natural language questions into structured queries (Liang et al., 2011; Cai and Yates, 2013; Kwiatkowski et al., 2013; Yao and Van Durme, 2014). Recent progresses include designing KB specific logical representation and parsing grammar (Berant et al., 2013), using distant supervision (Berant et al., 2013), utilizing paraphrase information (Fader et al., 2013; Berant and Liang, 2014), requiring little question-answer pairs (Reddy et al., 2014), and exploiting ideas from agenda-based parsing (Berant and Liang, 2015).

In contrast, another line of research tackles the problem by deep learning powered similarity matching. The core idea is to learn semantic representations of both the question and the knowledge from observed data, such that the correct supporting evidence will be the nearest neighbor of the question in the learned vector space. Thus, a main difference among several approaches lies in the neural networks proposed to represent questions and KB elements. While (Bordes et al., 2014b; Bordes et al., 2014a; Bordes et al., 2015; Yang et al., 2014) use relatively shallow embedding models to represent the question and knowledge, (Yih et al., 2014; Yih et al., 2015) employ a convolutional neural network (CNN) to produce the representation. In the latter case, both the question and the relation are treated as a sequence of letter-trigram patterns, and fed into two parameter shared CNNs to get their embeddings. What's more, instead of measuring the similarity between a question and an evidence triple with a single model as in (Bordes et al., 2015), (Yih et al., 2014; Yih et al., 2015) adopt a multi-stage approach. In each stage, one element of the triple is compared with the question to produce a partial similarity score by a dedicated model. Then, these partial scores are combined to generate the overall measurement.

Our proposed method is closely related to the second line of research, since neural models are employed to learn semantic representations. As in (Bordes et al., 2015; Yih et al., 2014), we focus on *single-fact questions*. However, we propose to use recurrent neural networks (RNN) to produce the question representation. More importantly, our method follows a probabilistic formulation, and our parameterization relies on factors other than similarity measurement.

Besides KB-based QA, our work is also loosely related to work using deep learning systems in QA tasks with free text evidences. For example, (Iyyer et al., 2014) focuses questions from the quiz bowl competition with recursive neural network. New architectures including memory networks (Weston et al., 2015), dynamic memory networks (Kumar et al., 2015), and more (Peng et al., 2015; Lee et al., 2015) have been explored under the `bAbI` syn-

thetic QA task (Weston et al., 2016). In addition, (Hermann et al., 2015) seeks to answer Cloze style questions based on news articles.

3 Overview

In this section, we formally formulate the problem of *single-fact question answering* with knowledge bases. A knowledge base $\mathcal{K}$ contains three components: a set of entities $\mathcal{E}$, a set of relations $\mathcal{R}$, and a set of facts $\mathcal{F} = \{\langle s, r, o\rangle\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where $s, o \in \mathcal{E}$ are the subject and object entities, and $r \in \mathcal{R}$ is a binary relation. $\mathrm{E}(r), \mathrm{E}(s)$ are the vector representations of a relation and an entity, respectively. $s \to r$ indicates that there exists some entity o such that $\langle s, r, o\rangle \in \mathcal{F}$. For single-fact questions, a common assumption is that the answer entity o and some triple $\langle s_i, r_k, o\rangle \in \mathcal{F}$ reside in the given knowledge base. The goal of our model is to find such subject s_i and relation r_k mentioned or implied in the question. Once found, a structured query (e.g. in SPARQL) can be constructed to retrieve the result entity.

3.1 Conditional Factoid Factorization

Given a question q, the joint conditional probability of subject-relation pairs $p(s, r|q)$ can be used to retrieve the answer using the exact inference defined by Eq. (1). However, since there can be millions of entities and thousands of relations in a knowledge base, it is less effective to model $p(s, r|q)$ directly. Instead, we propose a conditional factoid factorization,

$$p(s, r|q) = p(r|q) \cdot p(s|q, r) \tag{2}$$

and utilize two neural networks to parameterize each component, $p(r|q)$ and $p(s|q, r)$, respectively. Hence, our proposed method contains two phases: inferring the implied relation r from the question q, and inferring the mentioned subject entity s given the relation r and the question q.

There is an alternative factorization $p(s, r|q) = p(s|q) \cdot p(r|s, q)$. However, it is rather challenging to estimate $p(s|q)$ directly due to the vast amount of entities ($> 10^6$) in a KB. In comparison, our proposed factorization takes advantage of the relatively limited number of relations (on the order of thousands). What's more, by exploiting additional information from the candidate relation r, it's more feasible to model $p(s|q, r)$ than $p(s|q)$, leading to more robust estimation.

A key difference from prior multi-step approach is that our method do not assume any independence between the target subject and relation given a question, as does in the prior method (Yih et al., 2014). It proves effective in our experiments.

3.2 Inference via Focused Pruning

As defined by the Eq. (1), a solution needs to consider all available subject-relation pairs in the KB as candidates. With a large-scale KB, the number of candidates can be notoriously large, resulting in a extremely noisy candidate pool. We propose a method to prune the candidate space. The pruning is equivalent to a function that takes a KB $\mathcal{K}$ and a question q as input, and outputs a much limited set $\mathcal{C}$ of candidate subject-relation pairs.

$$\mathcal{H}(\mathcal{K}, q) \to \mathcal{C} \tag{3}$$

$\mathcal{C}_s$ and $\mathcal{C}_r$ are used to represent the subject and relation candidates, respectively.

The fundamental intuition for pruning is that the subject entity must be mentioned by some textual substring (*subject mention*) in the question. Thus, the candidate space can be restricted to entities whose name/alias matches an n-gram of the question, as in (Yih et al., 2014; Yih et al., 2015; Bordes et al., 2015). We refer to this straight-forward method as *N-Gram pruning*. By considering all n-grams, this approach usually achieves a high recall rate. However, the candidate pool is still noisy due to many non-subject-mention n-grams.

Our key idea is to reduce the noise by guiding the pruning method's attention to more probable parts of a question. An observation is that certain parts of a sentence are more likely to be the subject mention than others. For example, "*Harry Potter*" in "*Who created the character Harry Potter*" is more likely than "*the character*", "*character Harry*", etc. Specifically, our method employs a deep network to identify such focus segments in a question. This way, the candidate pool can be not only more compact, but also significantly less noisy.

Finally, combing the ideas of Eq.(2) and (3), we propose an approximate solution to the problem defined by Eq. (1)

$$\hat{s}, \hat{r} \approx \arg\max_{s,r \in \mathcal{C}} p(s|q, r)p(r|q) \tag{4}$$

4 Proposed CFO

In this section, we first review the gated recurrent unit (GRU), an RNN variant extensively used in

this work. Then, we describe the model parameterization of $p(r|q)$ and $p(s|q,r)$, and the focused pruning method in inference.

4.1 Review: Gated Recurrent Units

In this work we employ GRU (Cho et al., 2014) as the RNN structure. At time step t, a GRU computes its hidden state h_t using the following compound functions

$$z = \text{sigmoid}\left(W_{xz}x_t + W_{hz}h_{t-1} + b_z\right) \quad (5)$$

$$r = \text{sigmoid}\left(W_{xr}x_t + W_{hr}h_{t-1} + b_r\right) \quad (6)$$

$$\tilde{h} = \tanh\left(W_{xh}x_t + r \otimes W_{hh}h_{t-1} + b_h\right) \quad (7)$$

$$h_t = z \otimes h_{t-1} + (1 - z) \otimes \tilde{h} \quad (8)$$

where $W_{\{.\}}$, and $b_{\{.\}}$ are all trainable parameters. To better capture the context information on both sides, two GRUs with opposite directions can be combined to form a bidirectional GRU (BiGRU).

4.2 Model Parameterization

Relation network In this work, the probability of relations given a question, $p(r|q)$, is modeled by the following network

$$p_{\theta_r}(r|q) = \frac{\exp\big(v(r,q)\big)}{\sum_{r'}\exp\big(v(r',q)\big)} \quad (9)$$

where the *relation scoring function* $v(r,q)$ measures the similarity between the question and the relation

$$v(r,q) = f(q)^\top \mathrm{E}(r) \quad (10)$$

$\mathrm{E}(r)$ is the trainable embedding of the relation (randomly initialized in this work) and $f(q)$ computes the semantic question embedding. Specifically, the question q is represented as a sequence of tokens (potentially with unknown ones). Then, the question embedding model f consists of a word embedding layer to transform tokens into distributed representations, a two-layer BiGRU to capture the question semantics, and a linear layer to project the final hidden states of the BiGRU into the same vector space as $\mathrm{E}(r)$.

Subject network As introduced in section 3, the factor $p(s|q,r)$ models the fitness of a subject s appearing in the question q, given the main topic is about the relation r. Thus, two forces *a*) the raw context expressed by q, and *b*) the candidate topic described by r, jointly impact the fitness of the subject s. For simplicity, we use two additive terms to model the joint effect

$$p_{\theta_s}(s|q,r) = \frac{\exp\big(u(s,r,q)\big)}{\sum_{s'}\exp\big(u(s',r,q)\big)} \quad (11)$$

where $u(s,r,q)$ is the *subject scoring function*,

$$u(s,r,q) = g(q)^\top \mathrm{E}(s) + \alpha h(r,s) \quad (12)$$

$g(q)$ is another semantic question embedding, $\mathrm{E}(s)$ is a vector representation of a subject, $h(r,s)$ is the subject-relation score, and α is the weight parameter used to trade off the two sources.

Firstly, the *context score* $g(q)^\top \mathrm{E}(s)$ models the intrinsic plausibility that the subject s appears in the question q using vector space similarity. As $g(q)^\top \mathrm{E}(s)$ has the same form as equation (10), we let g adpot the same model structure as f. However, initializing $\mathrm{E}(s)$ randomly and training it with supervised signal, just like training $\mathrm{E}(r)$, is insufficient in practice — while a large-scale KB has millions of subjects, only thousands of question-triple pairs are available for training. To alleviate the problem, we seek two potential solutions: *a) pretrained* embeddings, and *b) type vector* representation.

The pretrained embedding approach utilizes unsupervised method to train entity embedings. In particular, we employ the `TransE` (Bordes et al., 2013), which trains the embedings of entities and relations by enforcing $\mathrm{E}(s) + \mathrm{E}(r) = \mathrm{E}(o)$ for every observed triple $(s,r,o) \in \mathcal{K}$. As there exists other improved variants (Gu et al., 2015), `TransE` scales the best when KB size grows.

Alternatively, type vector is a fixed (not trainable) vector representation of entities using type information. Since each entity in the KB has one or more predefined types, we can encode the entity as a vector (bag) of types. Each dimension of a type vector is either 1 or 0, indicating whether the entity is associated with a specific type or not. Thus, the dimensionality of a type vector is equal to the number of types in KB. Under this setting, with $\mathrm{E}(s)$ being a binary vector, let $g(q)$ be a continuous vector with arbitrary value range can be problematic. Therefore, when type vector is used as $\mathrm{E}(s)$, we add a sigmoid layer upon the final linear projection of g, squashing each element of $g(q)$ to the range $[0,1]$.

Compared to the first solution, type vector is fully based on the type profile of an entity, and requires no training. As a benefit, considerably

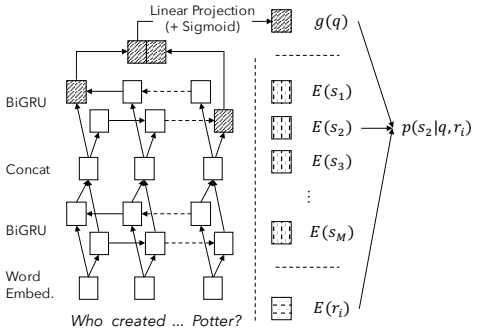

Figure 1: Overall structure of the subject network. Sigmoid layer is added only when type vector is used as $\mathrm{E}(s)$.

fewer parameters are needed. Also, given the type information is discriminative enough, using type vector will lead to easier generalization. However, containing only type information can be very restrictive.

In addition to the context score, we use the *subject-relation score* $h(r, s)$ to capture the compatibility that s and r show up together. Intuitively, for an entity to appear in a topic characterized by a relation, a necessary condition will be that the entity has the relation connected to it. Inspired by this structural regularity, in the simplest manner, we instantiate the idea with an indicator function,

$$h(r, s) = \mathbb{1}(s \to r) \tag{13}$$

As there exists other more sophisticated statistical parameterizations, the proposed approach is able to capture the core idea of the structural regularity without any parameter. Finally, putting two scores together, Fig.1 summarizes the overall structure of the subject network.

4.3 Focused Pruning

As discussed in section 3.2, N-Gram pruning is still subject to large amount of noise in inference due to many non-subject-mention n-grams. Motivated by this problem, we propose to reduce such noise by focusing on more probable candidates using a special-purpose sequence labeling network. Basically, a sequence labeling model is trained to tag some consecutive tokens as the subject mention. Following this idea, during inference, only the most probable n-gram predicted by the model will be retained, and then used as the subject mention to generate the candidate pool $\mathcal{C}$. Hence, we refer to this method as *focused pruning*. Formally, let $\mathcal{W}(q)$ be all the n-grams of the question q, $p(w|q)$ be the probability that the n-gram w is the subject mention of q, the focused pruning function $\mathcal{H}_s$ is defined as

$$\begin{aligned} \hat{w} &= \arg\max_{w \in \mathcal{W}(q)} p_\kappa(w|q) \\ \mathcal{C} &= \{(s, r) : \mathcal{M}(s, \hat{w}), s \to r\} \end{aligned} \tag{14}$$

where $\mathcal{M}(s, \hat{w})$ represents some predefined match between the subject s and the predicted subject mention $\hat{w}$. Intuitively, this pruning method resembles the human behavior of first identifying the subject mention with the help of context, and then using it as the key word to search the KB.

To illustrate the effectiveness of this idea, we parameterize $p_\kappa(w|q)$ with a general-purpose neural labeling model, which consists of a word embedding layer, two layers of BiGRU, and a linear-chain conditional random field (CRF). Thus, given a question q of length T, the score of a sequence label configuration $y \in \mathbb{R}^T$ is

$$s(y, q) = \sum_{t=1}^{T} \mathbf{H}(q)_{t,y_t} + \sum_{t=2}^{T} \mathbf{A}_{y_{t-1},y_t}$$

where $\mathbf{H}(q)$ is the hidden output of the top-layer BiGRU, $\mathbf{A}$ is the transition matrix possesed by the CRF, and $[\cdot]_{i,j}$ indicates the matrix element on row i collum j.

Finally, the match function $\mathcal{M}(s, \hat{w})$ is simply defined as either strict match between an alias of s and $\hat{w}$, or approximate match provided by the Freebase entity suggest API [1]. Note that more elaborative match function can further boost the performance, but we leave it for future work.

5 Parameter Estimation

In this section, we discuss the parameter estimation for the neural models presented in section 4.

With standard parameterization, the focused labeling model $p_\kappa(w|q)$ can be directly trained by maximum likelihood estimation (MLE) and back-propagation. Thus, we omit the discussion here, and refer readers to (Huang et al., 2015) for details. Also, we leave the problem of how to obtain the training data to section 6.

5.1 Decomposable Log-Likelihood

To estimate the parameters of $p_{\theta_r}(r|q)$ and $p_{\theta_s}(s|r, q)$, MLE can be utilized to maximize the empirical (log-)likelihood of subject-relation pairs

[1] The approximate match is used only when there is no strict match. The suggest API takes a string as input, and returns no more than 20 potentially matched entities.

given the associated question. Following this idea, let $\{s^{(i)}, r^{(i)}, q^{(i)}\}_{i=1}^{N}$ be the training dataset, the MLE solution takes the form

$$\theta^{\text{MLE}} = \arg\max_{\theta_r, \theta_s} \sum_{i=1}^{N} \Big(\log p_{\theta_r}(r^{(i)}|q^{(i)}) + \log p_{\theta_s}(s^{(i)}|r^{(i)}, q^{(i)}) \Big) \tag{15}$$

Note that there is no shared parameter between $p_{\theta_s}(s|q,r)$ and $p_{\theta_r}(r|q)$. [2] Therefore, the same solution can be reached by separately optimizing the two log terms, i.e.

$$\begin{aligned} \theta_r^{\text{MLE}} &= \arg\max_{\theta_r} \sum_{i=1}^{N} \log p_{\theta_r}(r^{(i)}|q^{(i)}) \\ \theta_s^{\text{MLE}} &= \arg\max_{\theta_s} \sum_{i=1}^{N} \log p_{\theta_s}(s^{(i)}|r^{(i)}, q^{(i)}) \end{aligned} \tag{16}$$

It is important to point out that the decomposability does not always hold. For example, when the parametric form of $h(s,r)$ depends on the embedding of r, the two terms will be coupled and joint optimization must be performed. From this perspective, the simple form of $h(s,r)$ also eases the training by inducing the decomposability.

5.2 Approximation with Negative Samples

As the two problems defined by equation (16) take the standard form of classification, theoretically, cross entropy can used as the training objective. However, computing the partition function is often intractable, especially for $p_{\theta_s}(s|r,q)$, since there can be millions of entities in the KB. Faced with this problem, classic solutions include contrastive estimation (Smith and Eisner, 2005), importance sampling approximation (Bengio et al., 2003), and hinge loss with negative samples (Collobert and Weston, 2008).

In this work, we utilize the hinge loss with negative samples as the training objective. Specifically, the loss function w.r.t θ_r has the form

$$\mathcal{L}(\theta_r) = \sum_{i=1}^{N} \sum_{j=1}^{M_r} \max\big[0, \gamma_r - v(r^{(i)}, q^{(i)}) + v(r^{(j)}, q^{(i)})\big] \tag{17}$$

where $r^{(j)}$ is one of the M_r negative samples (i.e. $s^{(i)} \not\rightarrow r^{(j)}$) randomly sampled from $\mathcal{R}$, and γ_r is the predefined margin. Similarly, the loss function w.r.t θ_s takes the form

$$\mathcal{L}(\theta_s) = \sum_{i=1}^{N} \sum_{j=1}^{M_s} \max\big[0, \gamma_s - u(s^{(i)}, r^{(i)}, q^{(i)}) + u(s^{(j)}, r^{(i)}, q^{(i)})\big] \tag{18}$$

Despite the negative sample based approximation, there is another practical difficulty when type vector is used as the subject representation. Specifically, computing the value of $u(s^{(j)}, r^{(i)}, q^{(i)})$ requires to query the KB for all types of each negative sample $s^{(j)}$. So, when M_s is large, the training can be extremely slow due to the limited bandwidth of KB query. Consequently, under the setting of type vector, we instead resort to the following type-wise binary cross-entropy loss

$$\tilde{\mathcal{L}}(\theta_s) = -\sum_{i=1}^{N} \sum_{k=1}^{K} \Big(\text{E}(s^{(i)})_k \log g(q^{(i)})_k + \big[1 - \text{E}(s^{(i)})_k\big] \log \big[1 - g(q^{(i)})_k\big] \Big) \tag{19}$$

where K is the total number of types, $g(q)_k$ and $\text{E}(s^{(i)})_k$ are the k-th element of $g(q)$ and $\text{E}(s^{(i)})$ respectively. Intuitively, with sigmoid squashed output, $g(q)$ can be regarded as K binary classifiers, one for each type. Hence, $g(q)_k$ reprents the predicted probability that the subject is associated with the k-th type.

6 Experiments

In this section, we conduct experiments to evaluate the proposed system empirically.

6.1 Dataset and Knowledge Base

We train and evaluate our method on the SIMPLEQUESTIONS dataset[3] — the largest question-triple dataset. It consists of 108,442 questions written in English by human annotators. Each question is paired with a subject-relation-object triple from Freebase. We follow the same splitting for training (70%), validation (10%) and testing (20%) as (Bordes et al., 2015). We use the same subset of Freebase (`FB5M`) as our knowledge base so that the results are directly comparable. It includes 4,904,397 entities, 7,523 relations, and 22,441,880 facts.

There are alternative datasets available, such as WebQuestions (Berant et al., 2013) and

[2] Word embeddings are not shared across models.

[3] `https://research.facebook.com/researchers/1543934539189348`

Free917 (Cai and Yates, 2013). However, these datasets are quite restricted in sample size — the former includes 5,810 samples (train + test) and the latter includes 917 ones. They are fewer than the number of relations in Freebase.

To train the focused labeling model, the information about whether a word is part of the subject mention is needed. We obtain such information by reverse linking from the ground-truth subject to its mention in the question. Given a question q corresponding to subject s, we match the name and aliases of s to all n-grams that can be generated from q. Once a match is found, we label the matched n-gram as the subject mention. In the case of multiple matches, only the longest matched n-gram is used as the correct one.

6.2 Evaluation and Baselines

For evaluation, we consider the same metric introduced in (Bordes et al., 2015), which takes the prediction as correct if both the subject and relation are correctly retrieved. Based on this metric, we compare CFO with a few baseline systems, which include both the Memory Network QA system (Bordes et al., 2015), and systems with alternative components and parameterizations from existing work (Yih et al., 2014; Yih et al., 2015). We did not compare with alternative subject networks because the only existing method (Yih et al., 2014) relies on unique textual name of each entity, which does not generally hold in knowledge bases (except in REVERB). Alternative approaches for pruning method, relation network, and entity representation are described below.

Pruning methods We consider two baseline methods previously used to prune the search space. The first baseline is the N-Gram pruning method introduced in Section 3, as it has been successfully used in previous work (Yih et al., 2014; Yih et al., 2015). Basically, it establishes the candidate pool by retaining subject-relation pairs whose subject can be linked to one of the n-grams generated from the question. The second one is N-Gram+, a revised version of the N-Gram pruning with additional heuristics (Bordes et al., 2015). Instead of considering all n-grams that can be linked to entities in KB, heuristics related to overlapping n-grams, stop words, interrogative pronouns, and so on are exploited to further shrink the n-gram pool. Accordingly, the search space is restricted to subject-relation pairs whose subject can be linked to one of the remaining n-grams after applying the heuristic filtering.

Relation scoring network We compare our proposed method with two previously used models. The first baseline is the embedding average model (Embed-AVG) used in (Bordes et al., 2014a; Bordes et al., 2014b; Bordes et al., 2015). Basically, it takes the element-wise average of the word embeddings of the question to be the question representation. The second one is the letter-tri-gram CNN (LTG-CNN) used in (Yih et al., 2014; Yih et al., 2015), where the question and relation are separately embedded into the vector space by two parameter shared LTG-CNNs. [4] In addition, (Yih et al., 2014; Yih et al., 2015) observed better performance of the LTG-CNN when substituting the subject mention with a special symbol. Naturally, this can be combined with the proposed focused labeling, since the latter is able to identify the potential subject mention in the question. So, we train another LTG-CNN with symbolized questions, which is denoted as LTG-CNN+. Note that this model is only tested when the focused labeling pruning is used.

Entity representation In section 4.2, we describe two possible ways to improve the vector representation of the subject, `TransE` pretrained embedding and type vectors. To evaluate their effectiveness, we also include this variation in the experiment, and compare their performance with randomly initialized entity embeddings.

6.3 Experiment Setting

During training, all word embeddings are initialized using the pretrained GloVe (Pennington et al., 2014), and then fine tuned in subsequent training. The word embedding dimension is set to 300, and the BiGRU hidden size 256. For pretraining the entity embeddings using `TransE` (see section 4.2), only triples included in `FB5M` are used. All other parameters are randomly initialized uniformly from $[-0.08, 0.08]$, following (Graves, 2013). Both hinge loss margins γ_s and γ_r are set to 0.1. Negative sampling sizes M_s and M_r are both 1024.

For optimization, parameters are trained using mini-batch AdaGrad (Duchi et al., 2011) with Momentum (Pham et al., 2015). Learning rates are

[4] In Freebase, each predefined relation has a *single* human-recognizable reference form, usually a sequence of words.

Pruning Method	Relation Network	Entity Representation		
		Random	Pretrain	Type Vec
Memory Network		62.9	63.9*	
N-Gram	Embed-AVG	39.4	42.2	50.9
	LTG-CNN	32.8	36.8	45.6
	BiGRU	43.7	46.7	55.7
N-Gram+	Embed-AVG	53.8	57.0	58.7
	LTG-CNN	46.3	50.9	56.0
	BiGRU	58.3	61.6	62.6
Focused Pruning	Embed-AVG	71.4	71.7	72.1
	LTG-CNN	67.6	67.9	68.6
	LTG-CNN+	70.2	70.4	71.1
	BiGRU	75.2	75.5	**75.7**

Table 1: Accuracy on SIMPLEQUESTIONS testing set. * indicates using ensembles. N-Gram+ uses additional heuristics. The proposed CFO (focused pruning + BiGRU + type vector) achieves the top accuracy.

tuned to be 0.001 for question embedding with type vector, 0.03 for LTG-CNN methods, and 0.02 for rest of the models. Momentum rate is set to 0.9 for all models, and the mini-batch size is 256. In addition, vertical dropout (Pham et al., 2014; Zaremba et al., 2014) is used to regularize all BiGRUs in our experiment. [5]

6.4 Results

Trained on 75,910 questions, our proposed model and baseline methods are evaluated on the testing set with 21,687 questions. Table 1 presents the accuracy of those methods. We evaluated all combinations of pruning methods, relation networks and entity representation schemes, as well as the result from memory network, as described in Section 6.1. CFO (focused pruning + BiGRU + type vector) achieves the best performance, outperforming all other methods by substantial margins.

By inspecting vertically within each cell in Table 1, for the same pruning methods and entity representation scheme, BiGRU based relation scoring network boosts the accuracy by 3.5 % to 4.8% compared to the second best alternative. This evidence suggests the superiority of RNN in capturing semantics of question utterances. Surprisingly, it turns out that Embed-AVG achieves better performance than the more complex LTG-CNN.

By inspecting Table 1 horizontally, type vector based representation constantly leads to better performance, especially when N-Gram pruning is used. It suggests that under sparse supervision, training high-quality distributed knowledge representations remains a challenging problem. That said, pretraining entity embeddings with `TransE` indeed gives better performance compared to random initialization, indicating the future potential of unsupervised methods in improving continuous knowledge representation.

In addition, all systems using our proposed focused pruning method outperform their counterparts with alternative pruning methods. Without using ensembles, CFO is already better than the memory network ensembles by 11.8%. It substantiates the general effectiveness of the focused pruning with subject labeling method regardless of other sub-modules.

6.5 Effectiveness of Pruning

According to the results in section 6.4, the focused pruning plays a critical role in achieving the best performance. To get a deeper understanding of its effectiveness, we analyze how the pruning methods affect the accuracy of the system. Due to space limit, we focus on systems with BiGRU as the relation scoring function and type vector as the entity representation.

Table 2 summarizes the recall — the percentage of pruned subject-relation candidates containing the answer — and the resulting accuracy. The single-subject case refers to the scenario that there is only one candidate entity in $\mathcal{C}_s$ (possibly with multiple relations), and the multi-subject case means there are multiple entities in $\mathcal{C}_s$. As the table shows, focused pruning achieves comparable recall rate to N-Gram pruning.[6] Given the state-of-the-art performance of sequence labeling systems, this result should not be surprising. Thus, the difference in performances entirely comes from their resulting accuracy. Notice that there exists a huge accuracy gap between the two cases. Essentially, in the single-candidate case, the system only need to identify the relation based on the more robust model $p_{\theta_r}(r|q)$. In contrast, under the multi-candidate case, the system also relies on $p_{\theta_s}(s|q,r)$, which has significantly more parameters to estimate, and thus is less robust. Consequently, by only focusing on the most probable sub-string, the proposed focused pruning produces much more single-candidate situations, leading to a better overall accuracy.

[5] For more details, source code is available at `http://zihangdai.github.io/cfo` for reference.

[6] Less than 3% of the recalled candidates rely on approximate matching in the focused pruning.

Pruning method	Pruning recall	Inference accuracy within the recalled		Overall accuracy
		Single-subject case	Multi-subject case	
N-Gram	94.8%	18 / 21 = 85.7%	12051 / 20533 = 58.7%	55.7%
N-Gram+	92.9%	126 / 138 = 91.3%	13460 / 20017 = 67.2%	62.6%
Focused pruning	94.9%	9925 / 10705 = 92.7%	6482 / 9876 = 65.6%	**75.7%**

Table 2: Comparison of different space pruning methods. N-Gram+ uses additional heuristics. Single- and multi-subject refers to the number of distinct subjects in candidates. The proposed focused pruning achieves best scores.

6.6 Additional Analysis

In the aforementioned experiments, we have kept the focused labeling model and the subject scoring network fixed. To further understand the importance and sensitivity of this specific model design, we investigate some variants of these two models.

Alternative focus with CRF RNN-CRF based models have achieved the state-of-the-art performance on various sequence labeling tasks (Huang et al., 2015; Lu et al., 2015). However, the labeling task we consider here is relatively unsophisticated in the sense that there are only two categories of labels - part of subject string (`SUB`) or not (`O`). Thus, it's worth investigating whether RNN (BiGRU in our case) is still a critical component when the task gets simple. Hence, we establish a CRF baseline which uses traditional features as input. Specifically, the model is trained with `Stanford CRF-NER` toolkit [7] on the same reversely linked labeling data (section 6.1). For evaluation, we directly compare the sentence level accuracy of these two models on the test portion of the labeling data. A sentence labeling is considered correct only when all tokens are correctly labeled. [8] It turns out the RNN-CRF achieves an accuracy of 95.5% while the accuracy of feature based CRF is only 91.2%. Based on the result, we conclude that BiGRU plays a crucial role in our focused pruning module.

Subject scoring with average embedding As discussed in section 4.2, the subject network g is chosen to be the same as f, mainly relying on a two-layer BiGRU to produce the semantic question embeding. Although it is a natural choice, it remains unclear whether the final performance is sensitive to this design. Motivated by this question, we substitute the BiGRU with an Embed-AVG model, and evalute the system performance.

[7] `http://nlp.stanford.edu/software/CRF-NER.shtml`

[8] As F-1 score is usually used as the metric for sequence labeling, sentence level accuracy is more informative here.

Relation Network	**Subject Network**	
	Embed-AVG	BiGRU
Embed-AVG	71.6	72.1
LTG-CNN	68.0	68.6
LTG-CNN+	70.4	71.1
BiGRU	75.4	75.7

Table 3: System performance with different subject network structures.

For this experiment, we always use focused pruning and type vector, but vary the structure of the relation scoring network to allow high-order interaction across models. The result is summarized in Table 3. Insepcting the table horizontally, when BiGRU is employed as the subject network, the accuracy is consistently higher regardless of relation network structures. However, the margin is quite narrow, especially compared to the effect of varying the relation network structure the same way. We suspect this difference reflects the fact that modeling $p(s|r, q)$ is intrinsically more challenging than modeling $p(r|q)$. It also suggests that learning smooth entity representations with good discriminative power remains an open problem.

7 Conclusion

In this paper, we propose CFO, a novel approach to single-fact question answering. We employ a conditional factoid factorization by inferring the target relation first and then the target subject associated with the candidate relations. To resolve the representation for millions of entities, we proposed type-vector scheme which requires no training. Our focused pruning largely reduces the candidate space without loss of recall rate, leading to significant improvement of overall accuracy. Compared with multiple baselines across three aspects, our method achieves the state-of-the-art accuracy on a 108k question dataset, the largest publicly available one. Future work could be extending the proposed method to handle more complex questions.

References

[Bengio et al.2003] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *The Journal of Machine Learning Research*, 3:1137–1155.

[Berant and Liang2014] Jonathan Berant and Percy Liang. 2014. Semantic parsing via paraphrasing. In *Proceedings of ACL*, volume 7, page 92.

[Berant and Liang2015] Jonathan Berant and Percy Liang. 2015. Imitation learning of agenda-based semantic parsers. *Transactions of the Association for Computational Linguistics*, 3:545–558.

[Berant et al.2013] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544.

[Bollacker et al.2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM.

[Bordes et al.2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795.

[Bordes et al.2014a] Antoine Bordes, Sumit Chopra, and Jason Weston. 2014a. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 615–620.

[Bordes et al.2014b] Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014b. Open question answering with weakly supervised embedding models. In *Machine Learning and Knowledge Discovery in Databases*, pages 165–180. Springer.

[Bordes et al.2015] Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston. 2015. Large-scale simple question answering with memory networks. *arXiv preprint arXiv:1506.02075*.

[Cai and Yates2013] Qingqing Cai and Alexander Yates. 2013. Large-scale semantic parsing via schema matching and lexicon extension. In *ACL (1)*, pages 423–433. Citeseer.

[Carlson et al.2010] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *AAAI*, volume 5, page 3.

[Cho et al.2014] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1724–1734.

[Collobert and Weston2008] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

[Duchi et al.2011] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

[Fader et al.2013] Anthony Fader, Luke S Zettlemoyer, and Oren Etzioni. 2013. Paraphrase-driven learning for open question answering. In *ACL (1)*, pages 1608–1618. Citeseer.

[Graves2013] Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

[Gu et al.2015] Kelvin Gu, John Miller, and Percy Liang. 2015. Traversing knowledge graphs in vector space. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 318–327.

[Hermann et al.2015] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1684–1692.

[Huang et al.2015] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991.

[Iyyer et al.2014] Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. 2014. A neural network for factoid question answering over paragraphs. In *Empirical Methods in Natural Language Processing*.

[Kumar et al.2015] Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2015. Ask me anything: Dynamic memory networks for natural language processing. *arXiv preprint arXiv:1506.07285*.

[Kwiatkowski et al.2013] Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke Zettlemoyer. 2013. Scaling semantic parsers with on-the-fly ontology matching. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

[Lee et al.2015] Moontae Lee, Xiaodong He, Wen-tau Yih, Jianfeng Gao, Li Deng, and Paul Smolensky. 2015. Reasoning in vector space: An exploratory study of question answering. *arXiv preprint arXiv:1511.06426*.

[Lehmann et al.2014] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, et al. 2014. Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 5:1–29.

[Liang et al.2011] Percy Liang, Michael I Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *Association for Computational Linguistics (ACL)*, pages 590–599.

[Liang et al.2013] Percy Liang, Michael I Jordan, and Dan Klein. 2013. Learning dependency-based compositional semantics. *Computational Linguistics*, 39(2):389–446.

[Lu et al.2015] Zefu Lu, Lei Li, and Wei Xu. 2015. Twisted recurrent network for named entity recognition. In *Bay Area Machine Learning Symposium*.

[Peng et al.2015] Baolin Peng, Zhengdong Lu, Hang Li, and Kam-Fai Wong. 2015. Towards neural network-based reasoning. *arXiv preprint arXiv:1508.05508*.

[Pennington et al.2014] 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*, pages 1532–1543.

[Pham et al.2014] Vu Pham, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour. 2014. Dropout improves recurrent neural networks for handwriting recognition. In *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, pages 285–290. IEEE.

[Pham et al.2015] Hieu Pham, Zihang Dai, and Lei Li. 2015. On optimization algorithms for recurrent networks with long short-term memory. In *Bay Area Machine Learning Symposium*.

[Reddy et al.2014] 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.

[Smith and Eisner2005] Noah A Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 354–362. Association for Computational Linguistics.

[Suchanek et al.2007] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM.

[Tang and Mooney2001] Lappoon R Tang and Raymond J Mooney. 2001. Using multiple clause constructors in inductive logic programming for semantic parsing. In *Machine Learning: ECML 2001*, pages 466–477. Springer.

[Weston et al.2015] Jason Weston, Sumit Chopra, and Antoine Bordes. 2015. Memory networks. In *International Conference on Learning Representations (ICLR2015)*.

[Weston et al.2016] Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. 2016. Towards ai-complete question answering: A set of prerequisite toy tasks. In *International Conference on Learning Representations (ICLR2016)*.

[Yang et al.2014] Min-Chul Yang, Nan Duan, Ming Zhou, and Hae-Chang Rim. 2014. Joint relational embeddings for knowledge-based question answering. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 645–650.

[Yao and Van Durme2014] Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *Proceedings of ACL*.

[Yih et al.2014] Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic parsing for single-relation question answering. In *Proceedings of ACL*.

[Yih et al.2015] 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 ACL*.

[Zaremba et al.2014] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *CoRR*, abs/1409.2329.

[Zelle and Mooney1996] John M Zelle and Raymond J Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the National Conference on Artificial Intelligence*, pages 1050–1055.