CASE-QA: Context and Syntax embeddings for Question Answering On Stack Overflow

Ezra Winston
Committee member: Graham Neubig
Advisor: William Cohen

Abstract

Question answering (QA) systems rely on both knowledge bases and unstructured text corpora. Domain-specific QA presents a unique challenge, since relevant knowledge bases are often lacking and unstructured text is difficult to query and parse. This project focuses on the QUASAR-S dataset (Dhingra et al., 2017) constructed from the community QA site Stack Overflow. QUASAR-S consists of Cloze-style questions about software entities and a large background corpus of community-generated posts, each tagged with relevant software entities. We incorporate the tag entities as context for the QA task and find that modeling co-occurrence of tags and answers in posts leads to significant accuracy gains. To this end, we propose CASE, a hybrid of an RNN language model and a tag-answer co-occurrence model which achieves state-of-the-art accuracy on the QUASAR-S dataset. We also find that this approach — modeling both question sentences and context-answer co-occurrence — is effective for other QA tasks. Using only language and co-occurrence modeling on the training set, CASE is competitive with the state-of-the-art method on the SPADES dataset (Bisk et al., 2016) which uses a knowledge base.

1 Introduction

Question answering (QA) is a long-standing goal of AI research. *Factoid QA* is the task of providing short answers — such as people, places, or dates — to questions posed in natural language. Systems for factoid QA have broadly fallen into two categories: those using knowledge-bases (KBs) and those using unstructured text. While KB approaches benefit from structured information, QA tasks which require domain-specific knowledge present a unique challenge since relevant knowledge bases are often lacking. Text-based approaches which query unstructured sources have improved greatly with recent advances in machine reading comprehension, but effective combination of search and reading systems is an active research challenge.

This project focuses on the QUASAR-S dataset (Dhingra et al., 2017) constructed from the community QA site Stack Overflow. QUASAR-S consists of Cloze-style (fill-in-the-blank) questions about software entities and a large background corpus of community-generated posts, each tagged with relevant software entities. To effectively answer these highly domain-specific questions requires deep understanding of the background corpus. One way to leverage the background posts corpus for QA is to train a language model of posts, creating training questions similar to the Cloze questions by treating entities in posts as answer entities. In this project, we find that additionally modeling co-occurrence of tags and answers in posts greatly aids in the QA task. For example, a post about Java and the Eclipse integrated development environment appears with tags *java*, *compilation*, and *java-7* and contains the sentence:

> *You can use the* ***eclipse*** *ide for the purpose of refactoring.*

We create a training question by treating *eclipse* as the *answer entity* and refer to the tags as the *context entities*. We use both the sentence q and the context entities c to predict the answer a, modeling $P(a|q,c)$.

This project proposes CASE, a hybrid of a recurrent neural network language model (RNN-LM) of question sentences $P(a|q)$ and a context-answer co-occurrence model of $P(a|c)$. Factoid questions can often be viewed as consisting of both a question sentence and one or more context entities. For example, the SPADES corpus (Bisk et al., 2016) contains questions about Freebase entities like *"USA has elected _blank_ , our* *first African-American president"* where we take *USA* to be the context entity and the desired answer entity is *Barack Obama*. We show that this view leads to a useful division of responsibility: the presence of the context model allows the he RNN LM to focus on the "type" of the answer entity based on question syntax.

This project makes the following original contributions:

- We propose CASE, a hybrid language/context model, and instantiate it using an RNN-LM and simple count-based co-occurrence context model.
- We show that CASE makes more effective use of background knowledge than both pure language modeling and *search-and-read* baselines, obtaining state-of-the-art performance on QUASAR-S.
- We demonstrate that on the SPADES dataset where no background text corpus is available, CASE still obtains results comparable to state-of-the-art knowledge-based methods, *without using a knowledge-base*. We then combine the co-occurrence counts with the best existing model to obtain a new state-of-the-art.
- Finally, we provide qualitative analysis of the entity embeddings produced by CASE, showing that they encode entity "type" information while ignoring semantic differences, which is of potential use for other tasks.

2 Background & Related Work

2.1 Problem Definition

We take an instance of the *QA with context* task to be a tuple (c, q, i) where $c = \{c_1, \dots, c_m\}$ is a set of one or more context entities, question sentence q has words $w_1, \dots, w_n$, and the answer a appears at index i, a.k.a. $w_i = a$. At test time the answer entity a is replaced with *_blank_* and the task is to identify it. That is, we wish to model $P(a|c, q_{\setminus w_i})$.

2.2 Question Answering

Research into both text-based and knowledge-based QA has recently centered on deep-learning approaches. For example, memory networks have proven an effective way to reason over KBs (e.g. (Bordes et al., 2015)). However, the relative sparsity of even the largest KBs has motivated a turn to unstructured text data such as Wikipedia articles. Such data is available in abundance but can prove challenging to retrieve and parse. Text-based approaches (e.g. Chen et al. (2017); Dhingra et al. (2017)) typically follow a *search-and-read* paradigm, involving a search stage, in which relevant documents are retrieved, and a reading stage, in which retrieved passages are read for the correct answer. Much research has focused on the reading stage, with many datasets (e.g. Rajpurkar et al. (2016)) developed for the reading comprehension task. Effectively trading off between query recall and reading accuracy is the subject of current research (Dhingra et al., 2017).

To our knowledge, little work has focused on incorporating background knowledge for QA via language modeling, although an RNN-LM is provided as a baseline on the QUASAR-S dataset (Dhingra et al., 2017). When applicable, this approach has the benefit of access to much larger training sets than either KB or search-and-read approaches, since it can be trained on natural-language sources that are orders of magnitude larger than existing QA training sets. In addition, the language-modeling approach does not depend on achieving the fine balance between query and reading systems required for search-and-read.

2.3 Language Modeling

Given a sequence S consisting of words $w_1, \dots, w_{k-1}$ (and sometimes words $w_{k+1}, \dots w_K$), the language modeling task is to model $P(w_k|S)$. Neural network language models such as those using LSTMs and GRUs have shown increasingly good performance (see Chung et al. (2014) for a comparison). Following (Dhingra et al., 2017), we adopt a BiGRU model for modeling the question sentence q.

RNN-LMs have trouble modeling long-range topical context as well as predicting rare words. We find that explicitly incorporating predictions based on context entities (e.g. tags in Stack Overflow, or Freebase entities in SPADES) is critical for the QA-with-context task, since the correct answer entity can be largely dictated by the context entities. Several approaches to incorporating long-range context in RNN-LMs have emerged and led to better language modeling performance. Following the terminology of Wang and Cho (2015), these either employ *early-fusion*, in which a context vector is concatenated with each RNN input

(Ghosh et al., 2016; Mikolov and Zweig, 2012), or late *late fusion*, in which a context vector is used as a bias before the output nonlinearity of the RNN cell (Wang and Cho, 2015).

We employ an approach most related to late-fusion, adding a context vector as a bias to the RNN output in logit space, prior to softmax. Related to our approach, Arthur et al. (2016) incorporate discrete lexicons into neural translation models by using them as a bias in the output softmax, finding that this compensates where neural translation models fail at translating rare but important words. Neubig and Dyer (2016) present a framework for hybridizing neural and n-gram language models, one instantiation of which involves neural interpolation between n-gram predictions and RNN-LM predictions. Also related to our approach is TopicRNN, a generative language model that combines a neural variational topic model over past words with an RNN language model of the current sentence (Dieng et al., 2016). Like CASE, TopicRNN injects long-range topical information by adding a topic bias in the output logit space.

3 CASE Models

We propose to use a language model $f(q, a) \propto P(a|q)$ together with a *context-entity* model of $g(c, a) \propto P(a|c)$ to model answer probabilities $P(a|c, q)$. We find that the conditional independence assumption

$$P(q, c|a) = P(q|c)P(c|a)$$

provides sufficient model complexity. This leads to the predictive distribution

$$\begin{aligned} P(a|q, c) &\propto P(a|q)P(a|c)/P(a) \\ &\propto f(q, a)g(c, a)/P(a). \end{aligned}$$

3.1 CASE-BiGRU-CC

Across all experiments we instantiate f as a bidirectional GRU network (BiGRU) used a baseline in Dhingra et al. (2017). Let $W_1 \in \mathbb{R}^{H\times V}$ be a word embedding matrix where V is the size of question word vocabulary $\mathcal{V}$ and H is the embedding dimension. Let $W_2 \in \mathbb{R}^{A\times 2H}$ be the output answer embedding matrix where A is the size of answer vocabulary $\mathcal{A}$. For predicting the entity at answer index i in question $q = w_1, \ldots, w_K$ we concatenate the forward and backward GRU outputs at that index:

$$\begin{aligned} x &= [W_1 w_1, \ldots, W_1 w_K] \\ h &= [fGRU(x)_{i-1}, bGRU(x)_{i+1}] \\ \log(f(q, \cdot)) &= W_2 h \end{aligned}$$

where the w_k are one-hot encoded and $fGRU(x)$ and $bGRU(x)$ are the sequential outputs of the forward and backward GRUs.

For the context model g we use simple co-occurrence counts calculated from the training set. Specifically, given context entities $c = \{c_1, \ldots, c_m\}$ we compute

$$g(c, a) = \text{avg}_i \frac{\#(a, c_i)}{\#(c_i)}.$$

In other words, for each context entity, we compute the empirical probability of co-occurrence with the answer entity, and then average over context entities in the context entity set.

Finally, answer predictions are

$$\begin{aligned} &P(\cdot|q, c) \\ &\quad = \text{softmax}\left(\log(f(q, \cdot)) - \log(g(c, \cdot)) - b\right) \\ &\quad \propto f(q, \cdot)g(c, \cdot)/\exp(b) \end{aligned}$$

where b is a learned bias.

3.2 Other Entity-Context Models

We also experimented with other choices of entity context model g. For example

- CASE-AE: $\log(g(c, \cdot)) = \text{avg}_i W c_i$, the **A**verage of context entity **E**mbeddings, where the c_i are one-hot encoded and W is a learned context entity embedding matrix.
- CASE-SE: **S**et **E**ncoding of the context entities, based on the self-attention set encoder suggested by Vinyals et al. (2015) for encoding unordered sets.

$$\begin{aligned} q_t &= GRU(q^*_{t-1}) \\ d_{i,t} &= \langle W c_i, q_t \rangle \\ a_{i,t} &= \text{softmax}(d_{\cdot,t}) \\ r_t &= \sum_i a_{i,t} c_i \\ q^*_t &= [q_t \; r_t] \\ \log(g(c, \cdot)) &= W q^*_m \end{aligned}$$

	QUASAR-S	SPADES
Training Qns	31,049	190,972*
Val Qns	3,174	4,763
Test Qns	3,139	9,309
Background Exs	17.8 mil†	-
Context Entities	44,375	53,961
Answer Entities	4,875	53,961‡

Table 1: Statistics of the QUASAR-S and SPADES datasets. *Each entity in the 79,247 original training questions is replaced to produce a new training question; †Each entity in the 26.6 mil. SO posts is replaced to produce a training example; ‡ While 1.8 million entities are present in the SPADES Freebase extract, we restrict prediction to entities appearing in the training questions.

where this process is repeated for $t = 0, \ldots, m$ steps, i.e. we take a number of self attention steps equal to the number of context entities.

As we report, these models failed to improve upon the pure BiGRU performance.

4 Experiments

4.1 Datasets

We conduct experiments on two datasets chosen to differ in both size and number of entities. Table 1 shows dataset statistics.

QUASAR-S (Dhingra et al., 2017): A large Cloze-style QA dataset created from the website Stack Overflow (SO), consisting of questions and background corpus in the computer programming domain. QUASAR-S has the unique feature of requiring deep domain expertise, making it unamenable to KB approaches. Non-expert humans achieve accuracy of 50% with open-book access to the same background corpus of posts, while experts achieved 46.8% in a closed-book setting.

The 37k Cloze questions are constructed from the definitions of SO tags by replacing occurrences of software entities with *_blank_*. The background corpus consists of 27 million sentences from the top 50 question and answer threads for each of 4,874 software entities. Each post is tagged with 1-5 tags. Figure 2 (top) shows an example question and relevant background sentences.

SPADES (Bisk et al., 2016): A set of 93k cloze-style questions constructed from sentences from ClueWeb09 (Gabrilovich et al., 2013) that contain two or more Freebase (Bollacker et al., 2008) entities linked by at least one relation path in Freebase. Unlike QUASAR-S, there is no background text corpus. In addition, no explicit tags are present. Instead, we take the non-answer entities in each question sentence (usually one) as the context entities. As in QUASAR-S we replace all occurrences of a context entity in a question with an "@context" token. Figure 2 (bottom) shows an example question.

4.2 Experimental Setup

Across all CASE-BiRNN-CC experiments we instantiate language model f as a single layer BiGRU with 128 hidden units following the baseline from Dhingra et al. (2017). For context model g we use co-occurrence counts as describe above. Training is conducted using a learning rate of 0.001 annealed by 50% after each epoch. We use the Adam (Kingma and Ba, 2014) optimizer with default hyperparameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1\text{e-}8$). Other BiGRU and learning parameters follow Dhingra et al. (2017), with 100-dimensional sentence word embeddings pretrained using skip-gram word2vec (Mikolov and Zweig, 2012).

QUASAR-S While the ultimate goal is to predict answers on the question set constructed from SO tag definitions, we first train on the large background post corpus. We create a training example for each occurrence of an answer entity in a post by replacing that entity with *_blank_* and treating it as the target answer. We use the one-to-five post tags as the context entities c. We prepend "@start" and "@end" tags to each post and replace all occurrences of topic tokens within the sentence with an "@context" token.

QUASAR-S questions have the tag being defined prepended to the definition sentence (see Figure 2 (top)). When evaluating on questions, we remove this tag from the sentence and use it as the context entity input to the model g.

Since the model is trained on posts and evaluated on the question set, we experimented with several transfer learning approaches for fine-tuning on the training question. We found that adding a post/question designator token to the beginning of training examples had no effect. Similar to the approach recommended for neural translation transfer learning in Chu et al. (2017), we first train on the large post corpus until convergence, then train on a 50/50 mix of training questions and posts. This procedure avoids overfitting to the training questions.

Question	java – java is a general-purpose object-oriented programming language designed to be used in conjunction with the java virtual-machine _______
Answer	**jvm**

Post tags	java, polymorphism, abstract, dynamic-binding
Post	this actually force the jvm to always check the run-time type of an object-reference

Figure 1: Example from Quasar-S: question from tag definition of tag "java" (top) and tagged post (bottom).

Sentence	Google acquired Nest which was founded in Palo Alto
Context entities	**Google**, **Palo Alto**
Question	Google acquired _______ which was founded in Palo Alto
Answer entity	**Nest**

Figure 2: Example from Spades.

We compare to the baselines reported in Dhingra et al. (2017) and also to the model CC consisting of only the co-occurrence counts model g, ignoring question sentences.

Spades We follow the same experimental procedure as for Quasar-S and use the same training/validation/test split as Das et al. (2017). For baselines we compare to the OnlyText, OnlyKB, and UniSchema models of Das et al. (2017). In addition to CASE-BiGRU-CC, we train hybrid models that add the co-occurrence counts as a bias to the output softmax of the OnlyKB and UniSchema models. For these models we use the code, parameters, and training procedures of UniSchema but train the model with the co-occurrence bias present. Finally, we compare to a model CC consisting of the co-occurrence model g only.

4.3 Results

Quasar-S Results and baselines are reported in Table 2. The fine-tuned CASE-BiGRU-CC obtains an accuracy of 45.2%, a gain of 11.6% over the best previously reported results of Dhingra et al. (2017) obtained by BiGRU (33.6%). Dhingra et al. (2017) also report performance of several search-and-read methods, the best of which uses the neural gated-attention (GA) reader. When the answer is present in a retrieved document, the GA reader identifies the correct answer 48.3% of the time, but 65% search accuracy limits overall accuracy to 31.6%. CASE-BiGRU-CC nearly matches the accuracy of the GA reader component alone. CASE-BiGRU-CC accuracy approaches that of human experts in a closed-book setting (46.8%), while falling 4.8% behind that of non-expert humans with search access to the back-

Method	**Val. Acc.**	**Test Acc.**
Human Performance		
Expert (CB)	0.468	-
Non-Expert (OB)	0.5	-
Language Models		
3-gram LM	0.148	0.153
4-gram LM	0.161	0.171
5-gram LM	0.165	0.174
BiGRU LM	0.345	0.336
Search + Read		
WD (SD)	0.100	0.107
MF-e (SD)	0.134	0.136
MF-i (SD)	0.159	0.159
GA (SD)	0.315	0.316
WD (LD)	0.082	0.093
MF-e (LD)	0.128	0.136
MF-i (LD)	0.159	0.159
GA (LD)*	0.318	0.321
New Models		
CC	0.128	0.139
BiGRU + ft	0.385	0.380
CASE-BiGRU-CC	0.413	0.413
CASE-BiGRU-CC + ft	**0.449**	**0.452**

Table 2: Performance comparison on Quasar-S. Results other than *New Models* are from Dhingra et al. (2017). ft: fine-tuning on questions; LD: long documents; SD: short documents; GA: gated-attention reader; MF-i, MF-e, WD: search-and-read methods using heuristics to extract answer from retrieved documents; OB: open-book; CB: closed book. See Dhingra et al. (2017) for details.

Method	Val. Acc.	Test Acc.
Bisk et al. (2016)	0.327	-
OnlyKB†	0.391	0.385
OnlyText†	0.253	0.266
Ensemble†	0.394	0.386
UniSchema†	0.411	0.399
CC	0.270	0.279
BiGRU	0.184	0.190
CaSE-BiGRU-CC	0.362	0.358
OnlyKB+CC	0.415	0.403
UniSchema+CC	**0.427**	**0.423**

Table 3: Performance comparison on Spades. †(Das et al., 2017)

ground post corpus (50.0%). We also note that the co-occurrence model alone (CC) gives a surprising 13.9% accuracy. Lastly, we find that fine-tuning on questions improves the performance of both the BiGRU and CaSE-BiGRU-CC by about 5%. We report negative results of the other context models below.

Spades Results and baselines are reported in Table 3. CaSE-BiGRU-CC, trained only on the question text, obtained better accuracy (35.8%) than both the BiGRU (19.9%) and the memory network OnlyText model of Das et al. (2017) which creates a knowledge base using training question text as facts. CaSE-BiGRU-CC performs nearly as well as the memory network OnlyKB model (38.6%) which uses Freebase facts and the UniSchema model which uses both text and Freebase facts. The co-occurrence only model CC obtains a surprising 27.9% accuracy. Using co-occurrence counts as a bias in the OnlyKB and UniSchema improve both by about 2.5% with the best model UniSchema+CC obtaining 42.3% accuracy.

4.4 Discussion

The inclusion of co-occurrence counts leads to significant gains on both dataset. This can be partially attributed to the performance of the CC model (co-occurrence count only) of 13.9% on Quasar-S and 27.9% on Spades, which can in turn be attributed to the Zipf's-law distribution of answer words. We posit that the surprising performances of CC on Spades is because sentences are restricted to correspond to some Freebase relation. This restriction means that (context entity, answer) pairs are frequently repeated.

On Quasar-S, the success of CaSE validates the idea that QA can take advantage of large text corpora with specialized domain knowledge, where no KB exists. We see a significant improvement over both the BiGRU and the search-and-read baselines. In the first case we attribute this to the fact that CaSE can effectively incorporate context entities while an RNN-LM cannot. In addition, the RNN in CaSE can focus more on syntactic/type information while the context/semantic information is handled by the entity context model g, which we explore further in Section 5. Interestingly, CaSE approaches GA-reader accuracy even when the correct answer is in context. This is likely due to training data requirements: while CaSE was trained directly on the 17 mil. post corpus, GA-reader was trained on only the 30k training questions, instead using the posts as the source for querying.

On the other hand, performance on Spades indicates that CaSE does not depend on a large corpus for language modeling. With co-occurrence counts capturing much of the information provided by a knowledge base, the language model makes a relatively smaller contribution than on Quasar-S. On Spades the language model contributes +7.9% accuracy over CC alone, compared with +31.3% on Quasar-S.

4.5 Negative Results

Other Context Models Neither of the two other entity context models for g, CaSE-AE and CaSE-SE, showed improvement over the BiGRU baseline. In both cases, we found that the model had difficulty learning context entity embeddings. We hypothesize that this is due in part to the highly non-uniform frequency of tags in the posts corpus, compared with the uniform distribution of tags in the test questions which come from definitions. This does not present a problem for the co-occurrence counts model, which does not need to learn context entity embeddings. Weighting training loss by inverse tag frequency may correct for this and is the subject of future work.

On Quasar-S we also experimented with other ways of incorporating context beyond the CaSE framework:

- CBiGRU: Similar to CLSTM (Ghosh et al., 2016). Instead of inputting embedding $W_1 w_i$ to the GRU we input $[Wc \;\; W_1 w_i]$ where Wc is an embedding for a single tag entity c. We train this model using only one tag for the context entity set c, so each post with m tags becomes m training examples with one tag each. Tag embeddings are initialized in the same way as vocab words, but are distinct from vocab word embeddings.

- BiGRU-PT: **P**repend **T**ags to the begin of each training post sentence, thus extending the length of the training post by m. The goal is to condition the GRU based on the contextual input.

- CASE-CC-Atten: Weight the contribution of each context entity co-occurrence using attention between the context entity embedding and the BiGRU output:

$$\begin{aligned} e_i &= \langle W_4 h, W_3 c_i \rangle \\ a_i &= \text{softmax}_i(e) \\ r &= \sum_i a_i g(c, \cdot)_i \\ P(\cdot|c, q) &= \text{softmax}(r + h - b) \end{aligned}$$

  where h is the output of the BiGRU. Tag embeddings are initialized as for CBiGRU.

We found that these alternative ways of incorporating context did not lead to improvement over the baseline BiGRU. The first two had trouble learning the context entity embeddings, as was the case with CASE-AE and CASE-SE. That BiGRU-PT did not show improved performance matches our intuition, since RNNs have trouble remembering context from the beginning of the sequence.

External data sources Attempts to train CASE on additional external data did not improve performance on either QUASAR-S or SPADES. On QUASAR-S we sought to augment the co-occurrence probabilities by using the Web Data Commons Web Table Corpora (Lehmberg et al., 2016), which includes 51 million English-language relational web tables from ClueWeb09. However, we found that only about 50,000 tables contained at least 2 pairs of SO software entities, and few of these tables were informational.

We also attempted to augment the training corpus for SPADES by incorporating sentences from Wikilinks (Singh et al., 2012), which consists of sentences from Clueweb09 that include hyperlinks to Wikipedia articles. Using this, we derived a link to Freebase and retained those sentences that had at least two linked entities. All in all, we augmented the original 79,247 SPADES training sentences with an additional 101,685 sentences. Comparing co-occurrence counts in this dataset to those in the SPADES training set showed that the two were distributed very differently. For example, given entity *Barack Obama*, entity *United States* co-occurred in 25% of SPADES training examples, but only 0.3% of Wikilinks sentences. We posit that this is again due to the restriction of the SPADES dataset to sentences with corresponding Freebase relations. Using Wikilinks co-occurrence counts performed much worse than SPADES training set co-occurrence counts (4.9% vs 27.0% acc.), and led to worse performance when combined with the BiGRU (24.4% vs 36.2% acc.).

Seed	CASE-BiGRU-CC	BiGRU
iphone	ipad	ios
sbt	gradle	intellij-idea
nginx	iis	.htaccess
excel	ms-word	xls
junit	rspec	testing
multiprocessing	parallel-processing	thread-safety
hadoop	mpi	hdfs

Table 4: Nearest neighbors in the CASE-BiGRU-CC and BiGRU output embedding space.

5 Analysis of Embeddings

We observe that by modeling context and question sentence separately, CASE factors entity representation into a semantic/contextual component given by context and a syntactic/type component given by the sentence. To assess the extent of this property we analyze the output entity embeddings learned by CASE-BiGRU-CC. To obtain (noisy) ground-truth types for SO entities, we link entities to Wikidata (Vrandečić and Krötzsch, 2014) via the links to Wikipedia in Stack Overflow tag definitions. We choose 20 groups of entities such as *Programming Languages* and *Network Protocols*. SPADES types are obtained from Freebase. Figure 3 shows that embeddings are well clustered by entity type.

To compare CASE-BiGRU-CC output embeddings to those of the BiGRU trained alone, we conduct two experiments. For both BiGRU and CASE-BiGRU-CC, we use output embeddings to predict type using 1-nearest-neighbor with cosine distance. Consistent with our expectations, CASE-BiGRU-CC embeddings obtain better accuracy (QUASAR-S: 63.3%, SPADES: 77.9%) that the BiGRU (QUASAR-S: 57.4%, SPADES:

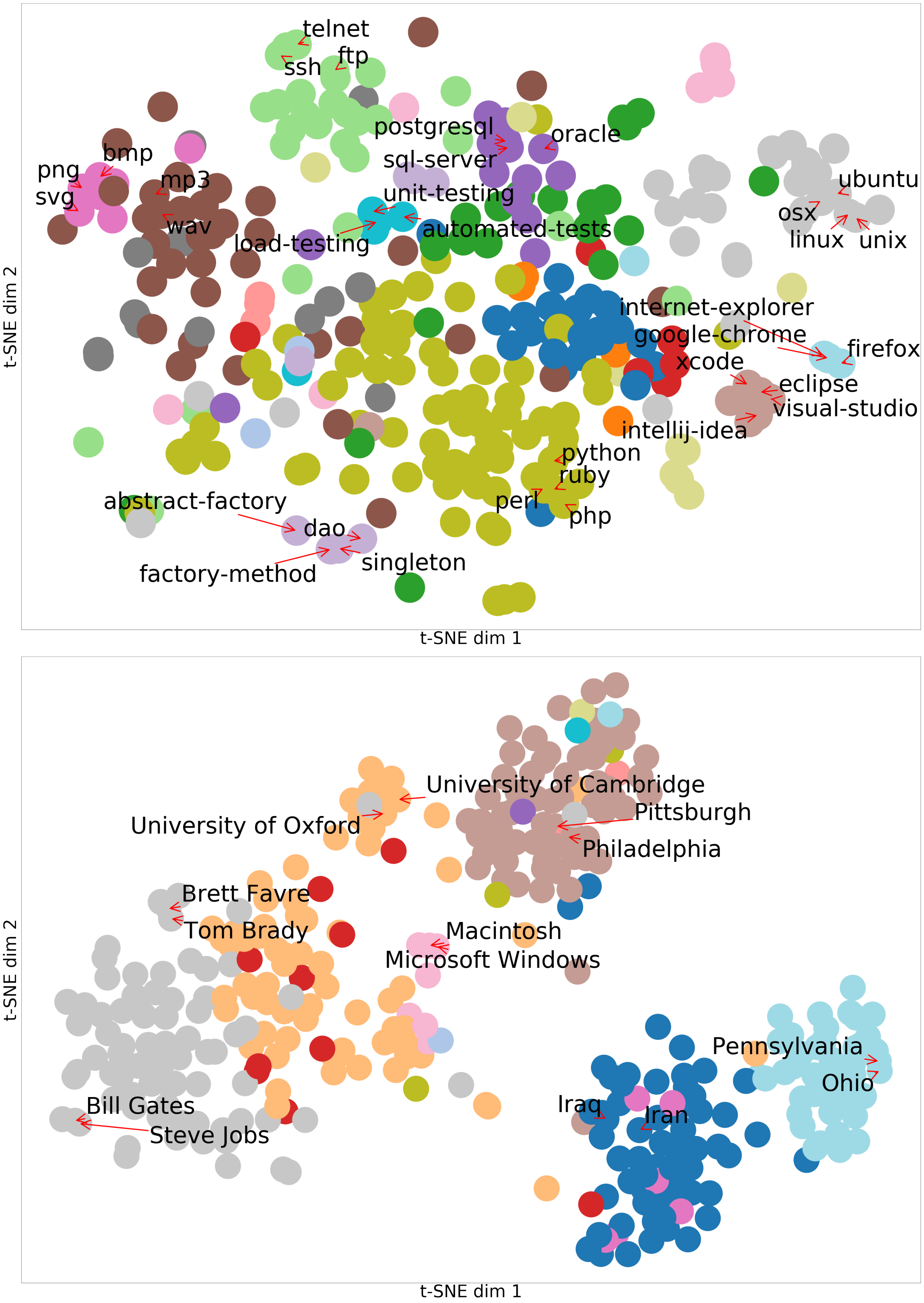

Figure 3: t-SNE (Maaten and Hinton, 2008) representation of output entity embeddings of CASE-BiGRU-CC on QUASAR-S (top) and SPADES (bottom). QUASAR-S entities are colored by their Wikidata type, SPADES entities by their Freebase type.

71.3%). Qualitatively, we observe several instances in which the nearest neighbors in CASE-BiGRU-CC embedding space are of the same type (e.g both build tools) while nearest neighbors in BiGRU embedding space may be only semantically related (e.g. a build tool and an IDE) (see Table 4).

6 Conclusions and Future Work

We demonstrated that combining a language model with a simple co-occurrence model of context entities leads to performance improvements on two Cloze-style QA tasks. CASE shows potential for domain-specific QA tasks such as QUASAR-S, where relevant knowledge bases are not available and search-and-read systems face difficulties. We see potential to incorporate other data sources into the context entity model, allowing for semi-structured data such as HTML web tables to be utilized. In addition, using more expressive models of context may improve performance. Finally, we showed that CASE embeddings encode type/syntax information. The application of these embeddings to other tasks warrants further investigation.

7 Acknowledgments

This work was funded by NSF under grant CCF-1414030 and by grants from Google.

References

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. *arXiv preprint arXiv:1606.02006* .

Yonatan Bisk, Siva Reddy, John Blitzer, Julia Hockenmaier, and Mark Steedman. 2016. Evaluating induced ccg parsers on grounded semantic parsing. *arXiv preprint arXiv:1609.09405* .

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*. ACM, New York, NY, USA, SIGMOD '08, pages 1247–1250.

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

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051* .

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of simple domain adaptation methods for neural machine translation. *arXiv preprint arXiv:1701.03214* .

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* .

Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. 2017. Question answering on knowledge bases and text using universal schema and memory networks. In *ACL*.

Bhuwan Dhingra, Kathryn Mazaitis, and William W Cohen. 2017. Quasar: Datasets for question answering by search and reading. *arXiv preprint arXiv:1707.03904* .

Adji B Dieng, Chong Wang, Jianfeng Gao, and John Paisley. 2016. Topicrnn: A recurrent neural network with long-range semantic dependency. *arXiv preprint arXiv:1611.01702* .

Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. Facc1: Freebase annotation of clueweb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0) .

Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, and Larry Heck. 2016. Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291* .

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. 2016. A large public corpus of web tables containing time and context metadata. In *Proceedings of the 25th International Conference Companion on World Wide Web*. International World Wide Web Conferences Steering Committee, pages 75–76.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research* 9(Nov):2579–2605.

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. *SLT* 12:234–239.

Graham Neubig and Chris Dyer. 2016. Generalizing and hybridizing count-based and neural language models. *arXiv preprint arXiv:1606.00499* .

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* .

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2012. Wikilinks: A large-scale cross-document coreference corpus labeled via links to wikipedia. *University of Massachusetts, Amherst, Tech. Rep. UM-CS-2012-015* .

Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2015. Order matters: Sequence to sequence for sets. *arXiv preprint arXiv:1511.06391* .

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

Tian Wang and Kyunghyun Cho. 2015. Larger-context language modelling. *arXiv preprint arXiv:1511.03729* .