International Journal of Computational Linguistics and Applications, Vol. 6, No. 2, 2015, pp. 105–116
Received 23/01/2015, Accepted 27/02/2015, Final 09/04/2015.
ISSN 0976-0962, http://ijcla.bahripublications.com

Simple, Fast Semantic Parsing with a Tensor Kernel

DAOUD CLARKE
University of Sussex, Falmer, Brighton, UK

ABSTRACT

We describe a simple approach to semantic parsing based on a tensor product kernel. We extract two feature vectors: one for the query and one for each candidate logical form. We then train a clasifier using the tensor product of the two vectors. Using very simple features for both, our system achieves an average F1 score of 40.1% on the WEBQUESTIONS dataset. This is comparable to more complex systems but is simpler to implement and runs faster.

1. INTRODUCTION

In recent years, the task of semantic parsing for querying large databases has been studied. This task differs from early work in semantic parsing in several ways:

- The databases being queried are typically several orders of magnitude larger, contain much more diverse content, and are less structured.
- In standard semantic parsing approaches, the aim is to learn a logical form to represent a query. In recent approaches the goal is to find the correct answer (entity or set of entities in the database), with learning a logical form a potential byproduct.
- Because of this, the datasets, which would have consisted of queries together with their corresponding logical forms, now

may consist of the queries together with the desired correct answer
- The datasets themselves are much larger, and cover a more diverse range of entities, however there may be a lot of overlap in the type of queries in the dataset.

We believe it is the last of these points that means that simple techniques such as the one we present can work surprisingly well. For example, the WEBQUESTIONS dataset contains 83 questions containing the term “currency”; of these 79 are asking what the currency of a particular country is. These 79 questions can be answered using the same logical form template, thus a system only has to see the term “currency”, and identify the correct country in the question to have a very good chance of getting the answer correct.

Knowing this on its own is not enough to build an effective system however. We still need to be able to somehow identify that it is this particular term in the query that is associated with this logical form. In this paper we demonstrate one way that this can be achieved. We build on the paraphrasing approach of [1] in that we use a fixed set of templates to generate a set of candidate logical forms to answer a given query and map each logical form to a natural language expression, its *canonical utterance*. Instead of using a complex paraphrasing model however, we use tensor kernels to find relationships between terms occuring in the query and in the canonical utterance. The virtue of our approach is in its simplicity, which both aids implementation and speeds up execution.

2. BACKGROUND

The task of semantic parsing initially focussed on fairly small problems, such as the GeoQuery dataset, which initially consisted of 250 queries [2] and was later extended to around 1000 queries [3]. Approaches to this task included inductive logic programming [2, 3], probabilistic grammar induction [4, 5], synchronous grammars [6] and induction of latent logical forms [7], the current state of the art on this type of dataset.

More recently, attention has focussed on answering queries in much larger domains, such as Freebase [8], which contains at the time of writing of around 2.7 billion facts. There are two datasets of queries for this database: FREE917 consisting of 917 questions annotated with logical forms [9], and WEBQUESTIONS which consists of 5,810 question-answer pairs, with no logical forms [10]. Approaches to this task include schema matching [9], inducing latent logical forms [10], application of paraphrasing techniques [1, 11], information extraction [12], learning low dimensional embeddings of words and knowledge base constituents [13] and application of logical reasoning in conjunction with statistical techniques [11]. Note that most of these approaches do not require annotated logical forms, and either induce logical forms when training using the given answers, or bypass them altogether.

2.1. *Semantic parsing via paraphrasing*

The PARASEMPRE system of [1] is based on the idea of generating a set of candidate logical forms from the query using a set of templates. For example, the query *Who did Brad Pitt play in Troy?* would generate the logical form

```
Character.(Actor.BraddPitt ⊓ Film.Troy)
```

as well as many incorrect logical forms. These are built by finding substrings of the query that approximately match Freebase entities and then applying relations that match the type of the entity. Given a logical form, a canonical utterance is generated, again using a set of rules, which depend on the syntactic type of the description of the entities.

To identify the most likely logical form given a query, a set of features are extracted from the query, logical form and canonical utterance:

```
what caused the asian currency crisis?
what countries use the euro as official
      currency?
what currency can you use in aruba?
what currency do i bring to cuba?
what currency do i need in cuba?
what currency do i need in egypt?
what currency do i take to turkey?
what currency do italy have?
what currency do mexico use?
what currency do the ukraine use?
what currency do they accept in kenya?
what currency do they use in qatar?
what currency do you use in costa rica?
what currency does brazil use?
what currency does greece use 2012?
what currency does greece use?
what currency does hungary have?
what currency does jamaica accept?
what currency does ontario canada use?
what currency does senegal use?
what currency does south africa have?
what currency does thailand accept?
what currency does thailand use?
what currency does the dominican republic?
what currency does turkey accept?
what currency in dominican republic should i
      bring?
what currency is best to take to dominican
      republic?
what currency is used in england 2012?
what currency is used in france before euro?
what currency is used in germany 2012?
what currency is used in hungary?
what currency is used in switzerland 2012?
what currency should i bring to italy?
what currency should i take to dubai?
what currency should i take to jamaica?
what currency should i take to mauritius?
what currency should you take to thailand?
what currency to take to side turkey?
what do you call russian currency?
what is australian currency?
what is currency in dominican republic?
what is currency in panama?
what is the best currency to take to egypt
      2013?
what is the currency in australia 2011?
what is the currency in croatia 2012?
what is the currency in england 2012?
what is the currency in france?
what is the currency in germany in 2010?
what is the currency in slovakia 2012?
what is the currency in the dominican republic
      2010?
what is the currency in the dominican republic
      called?
what is the currency in the republic of congo?
what is the currency name of brazil?
what is the currency name of china?
what is the currency of germany in 2010?
what is the currency of mexico called?
what is the currency of spain called?
what is the currency of sweden called?
what is the currency used in brazil?
what is the currency used in tunisia?
what is the local currency in the dominican
      republic?
what is the money currency in guatemala?
what is the money currency in italy?
what is the money currency in switzerland?
what is the name of currency used in spain?
what is the official currency in france?
what kind of currency do they use in thailand?
what kind of currency does cuba use?
what kind of currency does greece have?
what kind of currency does jamaica use?
what kind of currency to bring to mexico?
what money currency does canada use?
what the currency in argentina?
what type of currency does brazil use?
what type of currency does egypt have?
what type of currency does the us have?
what type of currency is used in puerto rico?
what type of currency is used in the united
      kingdom?
what type of currency should i take to mexico?
what's sweden's currency?
what's the egyptian currency?
which country has adopted the euro as its
      currency ( 1 point )?
which country uses euro as its main currency?
```

Figure 1. *Questions from the WEBQUESTIONS dataset containing the term "currency"*

- Features extracted from the logical form itself, such as the size of the denotation of a logical form, i.e. the number of results returned when evaluating the logical form as a query on the database. This is important, since many incorrect logical forms have denotation zero; this feature acts as a filter removing these.
- Features derived from an association model. This involves examining spans in the query and canonical utterance and looking for paraphrases between these spans. These

paraphrases are derived from a large paraphrase corpus and WordNet [14].

- Features derived from a vector space model built using Word2Vec [15].

In an analysis on the development set of WEBQUESTIONS, the authors showed that removing the vector space model lead to a small drop in performance, removing the asssociation model gave a larger drop, and removing both of these halved the performance score.

3. TENSOR KERNERLS FOR SEMANTIC PARSING

We know that simple patterns or occurrences in the query can be used to identify a correct logical form with high probability, as with the "currency" example. We still need some way of identifying these patterns and linking them up to appropriate logical forms. In this section we discuss one approach for doing this.

Our goal is to learn a mapping from queries to logical forms. One way of doing this to consider a fixed number of logical forms for each query sentence, and train a classifier to choose the best logical form given a sentence [1]. In order to use this approach, we need a single feature vector for each pair of queries and logical forms. Our proposal is to extract features for each query and logical form indepdendently, and to take their tensor product as the combined vector. Explicitly, let Q be the set of all possible queries and Λ be the set of all possible logical forms. For each query $q \in Q$ and logical form $\lambda \in \Lambda$ we represent the pair (q, λ) by the vector:

$$\phi(q, \lambda) = \phi_Q(q) \otimes \phi_\Lambda(\lambda)$$

where ϕ_Q and ϕ_Λ map queries and logical forms to a vector space, i.e. perform feature extraction.

Whilst this could potentially be a large space, note that we can use the kernel trick to avoid computing very large vectors, using a simple identity of dot products on tensor spaces:

$$\phi(q_1, \lambda_1) \cdot \phi(q_2, \lambda_2) = (\phi_Q(q_1) \cdot \phi_Q(q_2)) \, (\phi_\Lambda(\lambda_1) \cdot \phi_\Lambda(\lambda_2))$$

The advantage of using the tensor product is that it preserves all the information of the original vectors, allowing us to learn how features relating to queries map to features relating to logical forms.

More generally, instead of representing the query and logical form as vectors directly, this can be done implicitly using kernels. For example, we may use a
string kernel κ_1 on Q and a tree kernel κ_2 on Λ, then define the kernel $\kappa(q,\lambda) = \kappa_1(q)\ \kappa_2(\lambda)$ on $Q \times \Lambda$. This idea is closely related to the Schur product kernel [16].

It is worth noting at this point that, while what we really want is a one-to-one mapping from queries to logical forms, the classifier actually gives us a set of logical forms for each query: we simply ask it to classify each pair (q, λ). In a probabilistic approach, such as logistic regression, we can choose the λ for which the classifier gives the highest probability for (q, λ).

3.1. *Application to semantic parsing via paraphrasing*

There are clearly many ways we could map queries and logical forms to vectors. In this paper we will consider one simple approach in which we use unigrams as the features for both the query and the canonical utterance associated with the logical form. In this case, the tensor product of the vectors corresponds directly to the cartesian product of the unigrams derived from the query with those from the canonical utterance.

Recall that given two vector spaces U and V of dimensionality n and m, the tensor product space $U \otimes V$ has dimensionality nm. If we have bases for U and V, then we can construct a basis for $U \otimes V$. For each pair of basis vectors u and v in U and V respectively, we take a single basis vector $u \otimes v \in$

$U \otimes V$. In our case, the dimensions of U and V correspond to terms that can occur as unigram features in the query or canonical utterance respectively. Thus each basis vector of $U \otimes V$ corresponds to a pair of unigram features.

As an example from the WEBQUESTIONS dataset, consider the query, *What 5 countries border ethiopia?*, and the canonical utterance *The adjoins of ethiopia?*, whose associated logical form gives the correct answer. Then there will be a dimension in the tensor product for each pair of words; for example the dimensions associated with (*countries*, *adjoins*) and (*border*, *adjoins*), as well as less useful pairs such as (*5*, *ethiopia*) would all have non-zero values in the tensor product. Thus we are able to learn that if we see borders in the query, then a logical form whose canonical utterance contains the term adjoins is a likely candidate to answer the query.

4. EMPIRICAL EVALUATION

4.1. *Dataset*

We evaluated our system on the WEBQUESTIONS dataset [10]. This consists of 5,810 question-answer pairs. The questions were obtained by querying the Google Suggest API, and answers were obtained using Amazon Mechanical Turk. We used the standard train/test split supplied with the dataset, and used cross-validation on the training set for development purposes.

4.2. *Implementation*

We built our implementation on top of the PARASEMPRE system [1], and so our evaluation exactly matches theirs. Our implementation is freely available online.[1] We substituted the paraphrase system of PARASEMPRE with our tensor kernel-based system (i.e. we excluded features from both the association and vector space models), but we included the PARASEMPRE features derived from logical forms.

[1] Location witheld to preserve anonymity

To implement our tensor kernel of unigram features, we simply added all pairs of terms in the query and canonical utterance as features; in preliminary experiments we found that this was fast enough and we did not need to use the kernel trick, which could potentially provide further speed-ups. We did not implement any feature selection methods which may also help with efficiency.

For evaluation, we report the average of the F1 score measured on the set of entities returned by the logical form when evaluated on the database, when compared to the correct set of entities. This allows, for example, to get a non-zero score for returning a similar set of entities to the correct one. For example, if we return the set {Jaxon Bieber} as an answer to the query *Who is Justin Bieber's brother?* we allow a nonzero score (the correct answer according to the dataset is {Jazmyn Bieber, Jaxon Bieberg).

4.3. *Results*

Results are reported in Table 1. Our system achieves an average F1 score of 40.1%, compared to PARASEMPRE'S 39.9%. Our system runs faster however, due to the simpler method of generating features. Evaluating using PARASEMPRE on the development set took 22h31m; using the tensor kernel took 14h44m on a comparable machine.

Since we have adopted the logical form templates of PARASEMPRE, our upper bound or oracle F1 score is the same, 63% [1]. This is the score that would be obtained if we knew which was the best logical form out of all those generated. In contrast, Microsoft's DEEPQA has an oracle F1 score of 77.3% [11]; this could account for a large amount of the overall increase in their system. There is no reported oracle score for the Facebook system [13].

5. DISCUSSION

Table 2 shows the top unigram feature pairs after training on the WEBQUESTIONS training set. It is clear that, whilst there are some superfluous features that simply learn to replace a word with itself (for example *currency* with *currency*, there are obviously many useful features that would be nontrivial to identify accurately. There are also spurious ones such as the pair (*live*, *birthplace*); this is perhaps due to a large proportion of people who live in their birthplace.

Table 1. *Results on the WEBQUESTIONS dataset, together with results reported in the literature*

	Average F1 score
SEMPRE [10]	35.7
PARASEMPRE [1]	39.9
FACEBOOK [13]	41.8
DEEPQA [11]	45.3
Tensor kernel with unigrams	40.1

Table 2. *Top unigram pair features and their weights after training*

Feature	Weight	Feature	Weight
(currency, currency)	4.18	(name, who)	2.69
(parents, father)	3.46	(born, birth)	2.69
(die, death)	3.33	(influenced, influenced)	2.64
(religion, religion)	3.28	(live, birthplace)	2.63
(currency, used)	3.22	(country, birthplace)	2.62
(religions, religion)	3.11	(type, form)	2.62
(movies, film)	2.97	(do, profession)	2.60
(states, adjoins)	2.97	(died, death)	2.60
(timezone, zone)	2.95	(system, form)	2.60
(timezone, time)	2.94	(countries, country)	2.60
(speak, spoken)	2.91	(married, marry)	2.55
(currency, countries)	2.84	(language, language)	2.54
(money, currency)	2.82	(music, genres)	2.51
(capital, city)	2.77	(money, used)	2.47
(party, party)	2.75	(time, zone)	2.47
(nationality, country)	2.72	(wife, spouse)	2.46

In development, we found that ordering the training alphabetically by the text of the query lead to a large reduction in accuracy.[2] Ordering alphabetically when performing the split for

[2] We omit the values since they were performed on an earlier version of our code and are not comparable.

cross validation (instead of random ordering) means that a lot of queries on the same topic are grouped together, increasing the likelihood that a query on a topic seen at test time would not have been seen at training time. This validates our hypothesis that simple techniques work well because of the homogeneous nature of the dataset. We would argue that this does not invalidate the techniques however, as it is likely that real-world datasets also have this property.

It is a feature of our tensor product model that there is no direct interaction between the features from the query and those from the logical form. This is evidenced by the fact that the system has to $learn$ that the term $currency$ in the query maps to $currency$ in the canonical utterance. This hints at ways of improving over our current system. More interestingly, it also means that we are currently making very light use of the canonical utterance generation; in the canonical utterance, $currency$ could be replaced by any symbol and our system would learn the same relationship. This points at another route of investigation involving generating features for use in the tensor kernel directly from the logical form instead of via canonical utterances.

6. CONCLUSION

We have shown semantic parsing via paraphrasing using unigram features together with a tensor kernel performs comparably to more complex systems on the WEBQUESTIONS dataset. Our system is simpler to implement and runs faster.

In future work, as well as looking at more sophisticated feature inputs to the tensor kernel, we hope to work on improving the oracle F1 score.

REFERENCES

1. Berant, J. & Liang, P. 2014. Semantic parsing via paraphrasing. In proceedings of the *52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Baltimore,

Maryland, Association for Computational Linguistics (pp. 1415-1425).
2. Zelle, J. M. & Mooney, R. J. 1996. Learning to parse database queries using inductive logic programming. In proceedings of the *National Conference on Artificial Intelligence* (pp. 1050-1055).
3. Tang, L. R. & Mooney, R. J. 2001. Using multiple clause constructors in inductive logic programming for semantic parsing. In *Machine Learning: ECML 2001* (pp. 466-477), Springer.
4. Zettlemoyer, L. S. & Collins, M. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In proceedings of the *Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI2005)* (pp. 658-666), AUAI Press.
5. Kwiatkowksi, T., Zettlemoyer, L., Goldwater, S. & Steedman, M. 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In proceedings of the *2010 Conference on Empirical Methods in Natural Language Processing, Cambridge* (pp. 1223-1233), MA, Association for Computational Linguistics.
6. Wong, Y.W. & Mooney, R. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In proceedings of the *45th Annual Meeting of the Association of Computational Linguistics* (pp. 960-967), Prague, Czech Republic, Association for Computational Linguistics.
7. Liang, P., Jordan, M. & Klein, D. 2011. Learning dependency-based compositional semantics. In proceedings of the *49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (pp. 590-599), Portland, Oregon, USA, Association for Computational Linguistics.
8. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In proceedings of the *2008 ACM SIGMOD International Conference on Management of Data* (pp. 1247-1250), ACM.
9. Cai, Q. & Yates, A. 2013. Large-scale semantic parsing via schema matching and lexicon extension. In proceedings of the *51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 423-433), Sofia, Bulgaria, Association for Computational Linguistics.
10. Berant, J., Chou, A., Frostig, R. & Liang, P. Semantic parsing on Freebase from question-answer pairs. In proceedings of the *2013*

Conference on Empirical Methods in Natural Language Processing (pp. 1533-1544), Seattle, Washington, USA, Association for Computational Linguistics.

11. Wang, Z., Yan, S., Wang, H. & Huang, X. 2014. An Overview of Microsoft Deep QA System on Stanford WebQuestions Benchmark. Technical Report MSR-TR-2014-121.
12. Yao, X., Berant, J. & Durme, B. V. 2014. Freebase qa: Information extraction or semantic parsing? In *Workshop on Semantic Parsing*.
13. Bordes, A., Chopra, S. & Weston, J. 2014. Question answering with subgraph embeddings. In proceedings of the *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 615-620), Doha, Qatar, Association for Computational Linguistics.
14. Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.
15. Mikolov, T., Chen, K., Corrado, G. & Dean, J. 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
16. Shawe-Taylor, J. & Cristianini, N. 2004. Kernel methods for pattern analysis. Cambridge University Press, pub-CAMBRIDGE:adr.

DAOUD CLARKE
DEPARTMENT OF INFORMATICS,
UNIVERSITY OF SUSSEX,
FALMER, BRIGHTON, UK.
E-MAIL: <DAOUD.CLARKE@GMAIL.COM>