Sublexical Compositionality in Semantic Parsing

Diana Wan, Gunaa Arumugam Veerapandian

CS221 Project Progress Report,
Stanford University

jdwan@stanford.edu, avgunaa@stanford.edu

1. Introduction

Traditional sematic parsers have limitations such as requiring supervision to annotate data and only being able to operate on limited domains with few predicates. Previous work in Combinatorial Categorial Grammar usually relies on manually specified rules [1] or rules induced from annotated logical forms [2]. In this project, we are trying to implement a semantic parser that can scale up to Freebase without annotated logical forms.

2. Problem Description

Our goal is to train a sematic parser without annotated logical forms. We expect the output to be Michelle Obama if the input is Who is Barrack Obamas wife. This would require that we have a table where we can query Barrack Obama and spouse and get the output as Michelle Obama, and this would also require us to know that wife is a female spouse. We should also expect the input to be invalid if it is Who is Michelle Obamas wife because Michelle is matched to Obama and Michelle is female while Obama is male and also wife is a female spouse.

We will first develop a method to automatically build upon an existing predicate knowledge base to create mappings for more compositional predicates. Next we will modify existing alignment algorithms to utilize these compositional predicates and evaluate the effects on predicate alignment.

We will use Freebase as the existing single predicate knowledge database. We will use WordNet as the lexical database to provide cognitive synonym sets and conceptual relations in order to generate compositional predicate mappings. We will evaluate the new compositional predicate knowledge database on the same question-answer set as used in [3].

For finding the alignment, the input would be a list of text triples consisting of two arguments and a relation and a list of knowledge base triples with a similar structure. The output would be a mapping from these text relations to the logical composition of tables in the knowledge base.

3. Data collection

The data used in the project is obtained from two sources, one portion comes from freebase and the other is in the form of text triples.

Freebase Data : Freebase essentially contains a massive amount of data on various topics, we have gathered a subset of 14,873,062 data entries from coherent topics and categories like People, Business, Government and Organization. The reason for choosing this particular set of topics is to obtain closely related data which would have lot of entities overlapping. Each entry in this set is in the form of triples ($arg1 \rightarrow reln \rightarrow arg2$). For example, $fb : en.viswanathan\_anand \rightarrow fb : people.person.profession \rightarrow fb : en.chess\_master$.

Text Data : The text data consists of triples with text instead of freebase predicates. It is a subset of the data used in the original paper and it is again of the form ($arg1 \rightarrow reln \rightarrow arg2$). An example of this database would be $"Vishwanathan\ Anand" \rightarrow "profession\ of" \rightarrow "chess\ master"$. We have 3,000,000 data entries for the text triples.

For both the data sets, the arguments are referred to as entities and the relationship is referred to as the edge between two entities in later parts of the report. The basic goal of alignment using question answer pairs is to align "profession of" with fb:people.person.profession using certain techniques. For every such alignment we have a set of features like text frequency, KB frequency, intersection size, etc. Let us now look at the algorithm by which we can align possible formulas or predicates for different words.

4. Algorithm

Given two sets of triples, the knowledge base triples and the text based triples we learn the alignment by creating templates and looking for similar template pattern matches in the data.

4.1. Exact Match - Baseline

Baseline for the problem we are trying to solve uses only exact match between the text relation and the freebase relation to determine the alignments between formula and lexeme. For example, "fb:location.statistical_region.population" aligns to "population" since "population" is a part of the freebase relationship string. The method we use for testing the baseline and other models initially is using examples generated manually.

4.2. Template Based Model

In our algorithm, we use templates to achieve improvement over the baseline. The triples can be interpreted as a graph with entities as vertices and relationships as edges. We define a "template" as a pattern of the formula which would be used to check for matches in the graph. We will have many formulas from the freebase data for a particular template. Based on the intersection between the entity pairs in both the freebase triples and the text triples we include the formula to the alignment. Each (formula, relationship) pair has a list of feature values like text frequency, KB frequency, intersection size which is gathered from the data. Let us look at algorithm to handle certain templates and examples.

The various templates under consideration are:

1. **Template 1** : $X \rightarrow Y$
   Here X and Y stand for two entities and the edge between them represents the relationship between the entities. We need to find matches in both the Freebase data and the Text data. This template represent a straight forward alignment with between two relationships based on the intersection. For example: freebase data $fb : en.barrack_obama \rightarrow fb : people.person.place_of_birth \rightarrow fb : en.honolulu$ and text triple $"Barrack\ Obama" \rightarrow "born\ in" \rightarrow "honolulu"$ will result in a successful match and would result in the following alignment entry:

   formula = fb:people.person.place_of_birth
   lexeme = "born in"
   source = "ALIGNMENT"
   features = {FB_typed_size :16184.0, Intersection_size_typed:3.0, "NL-size":5.0}

2. **Template 2** : $X \rightarrow Mediator \rightarrow Y$
   Now we have entities X, Y and two edges which stand for two relationships in the database. Along with X and Y we see the inclusion of a new vertex Mediator. Mediator is a special Compound Value Type (CVT) node marked in the freebase data in a unique way. Mediators connect multiple entities together following a similar relationship. (Example, (i) the population of a country X in year 2010 was Y (ii) X was married to Y from t1 to t2) Similar to the previous template algorithm used, we match the pattern obtained using the data and include the alignment to our final output if the intersection is non empty. This would give us the ability to answer more queries.

3. **Template 3** :
$$\begin{tikzcd} X \arrow[r] \arrow[d] & Y \\ C & \end{tikzcd}$$
   X, Y are the main entities and C is like a constant with small domain (for example : gender = {male, female}) to which the entity X has a certain relationship with. For example, if we have a question which asks ”Who is the wife of Michelle Obama?” using the first template we would get an erroneous reply ”Barrack Obama”. So to the lexeme ”wife of” we also need to attach another relationship which refers to the gender of the entity. The new alignment would be

   formula =
   ”(lambda x (fb:people.person.spouse (fb:people.person.gender.female(var x))))”
   lexeme = ”wife of”
   source = ”ALIGNMENT”
   features = {FB_typed_size :aaa, Intersection_size_typed:bbb, ”NL-size”:ccc}

   The algorithm used to align the formula is still similar to the ones discussed above but we now use a unique lambda function which would represent the formula. This is yet to be implemented and tested.

   A few more templates which would be our targets for the end project are discussed with an example.

4. **Template 4** : $X \rightarrow Y \rightarrow Z$
   This template has 3 entities and can be used for aligning grandfather relationships. In our freebase data we have parent relationship but we do not have grandparent relationship. So grandparent can be formulated using lambda calculus as (lambda x (fb:people.person.parent (fb:people.person.parent(var x))))

5. **Template 5** :
$$\begin{tikzcd} X & Mediator \arrow[l] \arrow[r] \arrow[d] & Y \\ & Z & \end{tikzcd}$$
   An example for this template would be the following: If we want to find President of a country, we do not have a direct table in freebase which provides such information. Although data regarding important government officials is provided.

We have seen each of the templates with algorithms and examples. The modeling of features in the alignment entries have to be engineered separately in future to obtain good result with the testing.

5. Experiments and Observations

We obtained the alignments for the baseline using direct string match approach. For testing the baseline and oracle we created a set of examples which would fall under the range of the various templates discussed above. Using these test examples and the alignment we can estimate the accuracy of alignment of a formula to a lexeme.

For the baseline, which uses the exact match algorithm very few of the examples would actually have an alignment with a formula. Out of around 20 example text relationships 3 matches were correctly obtained. The percentage of match is 15% for the baseline. For the oracle the alignment was done manually using all templates we have described earlier. We obtained around 18 matches 90% accuracy on our simple test set. We hope to improve this value in future tests and also create a better testing algorithm using already available questions.

6. Future work

- Implement algorithms for each template mentioned above and use the new alignments for testing the model.
- Come up with intuitive template which would allow us to expand our scope and parse more sentences effectively.
- Feature engineer the various values in the output alignment and get good scores.
- Testing on the set of available webquestions would give us a good uniform way to analyse the alignment rather than coming up with our own test examples manually.

7. References

[1] J. Krishnamurthy and T. Mitchell. 2012. Weakly supervised training of semantic parsers. In Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL), pages 754765.

[2] T. Kwiatkowski, L. Zettlemoyer, S. Goldwater, and M. Steedman. 2010. Inducing probabilistic CCG grammars from logical form with higher-order unification. In Empirical Methods in Natural Language Processing (EMNLP), pages 12231233.

[3] J. Berant, A. Chou, R. Frostig, P. Liang, Semantic Parsing on Freebase from QuestionAnswer Pairs, ACL, 2013.