Sublexical Compositionality in Semantic Parsing

Diana Wan, Gunaa Arumugam Veerapandian

CS221 Project Progress Report,
Stanford University
jdwan@stanford.edu, avgunaa@stanford.edu

1. Introduction

2. Problem Description

3. Aligning Predicates from Question Answer Pairs

4. 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. 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.

5. 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.

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 some templates and their examples which we use to find patterns in the data.

The various templates under consideration are:

1. Case 1 **(Baseline)**:$X \rightarrow Y$
   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}

   Baseline for the problem we are trying to solve uses only this template to determine the alignments between formula and lexeme. The method we use for testing the baseline and other models initially is using examples generated manually. Other templates we will be using for improving the baseline model is given henceforth.

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. $X \rightarrow Y$
   $\downarrow$
   C
   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. $X \rightarrow Y \rightarrow Z$ 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. $X \leftarrow Mediator \rightarrow Y$
   . $\downarrow$
   . Z
   An example for this template would be the following: Consider the

We have seen each of the templates with algorithms and examples to handle the cases. We also mentioned about the features in each alignment entry, they have to be engineered separately in future to obtain good result with the testing.

6. Experiments and Observations

For testing the baseline and oracle we creating

7. Algorithm

8. Future work

9. References