Analogical Inference over a Common Sense Database

Thomas Lin

MIT Media Laboratory
410 Memorial Drive
Cambridge, MA 02139
tlin@mit.edu

Introduction

This paper shows that by applying analogical inference techniques to a large natural language common sense database, we can generate new, plausible common sense facts. Two systems that do this are described. Being able to generate new facts in this manner allows quick augmentation of the common sense database.

The role of analogical inference in common sense reasoning has been discussed before (Carbonell 1983), but only recently have large common sense databases become publicly available. We used the Open Mind Common Sense (OMCS) database, which contains several hundred thousand English common sense statements (Singh 2002).

The OMCS database is built by internet users who are prompted to enter common sense facts at a website. The reason we use the OMCS database instead of the CYC common sense database (Lenat and Guha 1994) is because this analogical inference work is concurrently being integrated into OMCS's data collection mechanism.

The idea is that an internet user can enter a fact at the OMCS website, then the system uses techniques described by this paper to respond with 30 plausible deductions based off the fact. Then, the user decides whether each deduction is true or false. True deductions become additional database facts, while false ones become negative expertise. Doing this allows for much faster data input by users.

To carry out the analogical inference, we first represent the OMCS database as a set of concepts and a set of relations. A concept is a noun phrase or an adjective. A relation looks like “A ? is for playing ?”. Each original sentence becomes a relation after its concepts are replaced by question marks.

Analogies over Concepts and Relations

The first system finds “analogies over concepts and relations.” It takes a sentence as input, and outputs a list of inferences.

Analogy over concepts starts by finding all other relations connecting an original sentence's set of concepts. It then finds all other sets of concepts that these relations connect. It substitutes each new set of concepts into the original relation to form deductions. Here is an example:

1. A user enters: “A mother can have a baby.”
2. The system parses this into “A ? can have a ?” with concepts “mother” and “baby”
3. The system finds all other relations in the database for “mother” and “baby,” such as “A ? will feed her ?”
4. Then it finds all other sets of concepts connected by each such relation. For instance, “girl” and “small dog” from “A girl will feed her small dog.”
5. It substitutes each new set of concepts back into the original relation. In our example, this gives “A girl can have a small dog.” This new sentence is added to the inferences list.

Analogy over relations starts by finding all other sets of concepts connected by the relation present in the original sentence. It then finds all other relations connecting each such set of concepts and substitutes the concepts from the original sentence into the new relations. Here is an example:

1. A user enters: "A mother can have a baby."
2. The program parses this as "A ? can have a ?"
3. Now it finds all other sets of concepts that are connected by this relation, such as "A child can have a goldfish."
4. For each such set of concepts, it finds the other relations that connect the set. For example, "child" and "goldfish" are also connected by "A child can take care of a goldfish."
5. This has the pattern "A ? can take care of a ?"
6. Each of these patterns is then filled with the original concepts. Here, we get "A mother can take care of a baby." This new sentence is added to the inferences list.

This system has used “Hawks eat rabbits” to deduce “There is more rabbits than there are hawks” by relating (hawks, rabbits) to (cows, grass) and finding “There is more grass than there are cows.”

Analogies as Inference Rules

The second system does analogical inference by generating a list of inference rules. First, the original database is organized into a graph where the nodes are concepts and the edges are relations. Then, it finds all cycles in the graph where three 2-part relations connect three concepts. (A 2-part relation is a relation over exactly 2 variables). This could look like,

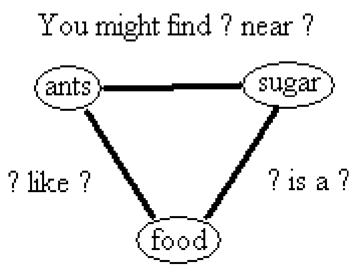

Figure 1: In this graph, "ants," "sugar," and "food" are in a cycle. The three original sentences are: "You might find ants near sugar," "Ants like food," and "Sugar is a food."

Each cycle is an inference rule. The rule here is: "(?a like ?b) (?c is a ?b) (You might find ?a near ?b)." For each rule, the system finds all other places in the graph where any two of the three elements in the inference rule are instantiated. It then forms what a third element needs to look like to be consistent with the first two, and presents it as an inference. Inference rules with more occurrences in the database are "better."

The input to the system is a sentence, and the output is all inferences that can be made with it. If the user enters "Bats like darkness" and "You might find bats near cave interiors" is already in the database, then the system matches: ?a = "bats," ?b = "darkness," and ?c = "cave interiors" to deduce "Cave interiors is a darkness" from the earlier example. This expresses a new idea that is not in the original database.

This system has used "Seeing requires that your eyes are open" to deduce "If you want to read a newspaper then you should have eyes."

Analysis

"Analogy over concepts and relations" is O(# of relations that connect any set of concepts) x O(# of concept sets that are connected by any relation) running time. "Analogies as inference rules" is O(# of inference rules that include the user's sentence's relation) x O(# of concept sets that are connected by any relation) running time. Both systems usually run within a second on a 800 MHz machine with 512 MB RAM, although "analogies as inference rules" take several seconds for certain input.

To test the validity of the inferences, 40 random sentences from the original database that had 2-part relations were used as input to both systems, and we analyzed the results.

55% of sentences produced output from "analogies over concepts and relations." 25% of overall sentences had matches from "analogy over concepts," while 40% had matches from "analogy over relations." The average sentence with matches from "analogy over concepts" had 66% of its results express new, true statements. The average sentence with matches from "analogy over relations" had 64.8% of its results express new, true statements.

67.5% of sentences had results from analogy with inference rules. The average such sentence had 35.6% of its results express new, true statements.

Sometimes the systems will make analogies that do not fit well. The probability of meaningful output can be increased by using systems of negative expertise, probability, and plausibility.

Conclusion

Two new analogical inference systems have been developed, implemented, and tested. The first one works by matching and substituting concepts and relations. The second one works by automatically generating a list of inference rules, then using these rules to make deductions.

The systems described in this paper work over large, noisy databases, not just toy examples. They will output deductions when fed new or existing sentences, and people can manually identify good and bad deductions to quickly augment the original database.

I would like to thank Push Singh for his guidance and encouragement with this work.

References

Carbonell, J. G. and Minton, S. 1983. Metaphor and Common-Sense Reasoning, Technical Report, CMU-CS-110, Carnegie-Mellon University.

Lenat, D. B. and R. V. Guha. 1994. Enabling Agents to Work Together. *Communications of the ACM* 37, no. 7.

Singh, P. 2002. The Public Acquisition of Commonsense Knowledge. In *Proceedings of AAAI Spring Symposium.* Menlo Park, Calif.: American Association of Artificial Intelligence, Inc.