Semantic Indexing

Anant Bhardwaj ANANTB@CSAIL.MIT.EDU
Ekaterina Ob'yedkova KATYA_@MIT.EDU
Ravi Netravali RAVINET@MIT.EDU

1. Introduction

Majority of information retrieval and web search use the inverted index as the backbone for their keyword based search. Unfortunately, inverted index fail to capture the semantic of the language and thus keyword based approach can only go as far as giving the relevant hits. We propose semantic indexing scheme which can encode natural languages without losing semantic relationships.

The proposed semantic encoding would involve parsing of each sentence, deriving its semantic model and storing the encoded representation. Matthew Fay and Jodyann Coley in his work SemantcSQL have attempted to encode natural language in SQL. While SemanticSQL can encode simple sentences, it fails to encode and accurately recover the semantic information from the stored representation for complex sentences. We aim to build a model that can derive semantic structure from complex sentences and a new schema which can accurately store the derived semantic model. In the following paragraph we illustrate different cases their system currently can not handle and propose a new semantic model which incorporates them.

1. **Probabilistic Parsing**
   Our system would incorporate a probabilistic sentence parser which would allow the system to intelligently identify the part-of-speech of previously unknown words and appropriately encode them.

2. **Multiple Adverbs and Prepositional Phrases**
   The current implementation only allows a single occurrence of location, adverb, and instrument phrases. Our system would allow sentences such as *John slept under the tree in the park* to be understood, and the question “Where did John sleep?” would return *under the tree in the park'*.

3. **Noun Phrases with Relative Classes, and Time**
   (a) Show me all the states with population greater than 1 million.
   (b) Show me all the widgets that were selected yesterday.

4. **Either-Or and Neither-Nor**
   (a) Josh ate either bread or apple (exclusive)
   (b) Josh didn't eat either bread or apple (negation, inclusive)
   (c) Josh ate neither bread nor apple (inclusive).

5. **Intersective vs. Non-Intersective Adjectives**
   Intersective Adjectives:
   (a) The car is a red Volkswagen.
   (b) Grace is a hairy brown dog.
   (c) We drove up yet another serpentine mountain road.

   Non-Intersective Adjectives:
   (a) Viktor is a former Catholic.
   (b) The alleged thief has arrived in court.

6. **Complex Causality**
   The current implementation only allows a primitive causality of 'if then' kind. Expanding it to more complex causality mapping would allow us handle complex causality.
   (a) Anna told Alex to return the chair because it was broken.
   (b) They moved here because of the work.
   (c) Because John is tired we decided to stay at home.
   (d) We can leave after the rain stops.
   (e) If Mary finishes her homework and the weather is good then Mary will go for a walk.

   Relating events based on time:
   (a) Did John come home after the fire alarm went off?

7. **Conjunctions and Disjunctions**
   (a) Depending on the weather, John will go for a walk or watch a movie.

8. **Quantifiers**
   For instance, we would be able to answer questions like:

(a) *How many times have John watched the movie?*
(b) *How much milk have John got in the fridge?*
(c) *Is there any milk in the fridge?*

We will also develop an infrastructure which can store the encoded semantic model for the types of sentences described above. This infrastructure would allow natural language query through our web interface and a REST Query interface so that other applications, regardless of the programming language they use, can build upon our system.

2. Schema

Events
id

Actions
id
action

Objects
id
object

Adjectives
id
adjective

Adverbs
id
adverb

Tenses
id
tense

Event-Actions
id
event_id
action_id
tense_id

Event-Agents
id
event_id
agent_id(object_id)

Event-Patients
id
event_id
patient_id(object_id)

Event-Instruments
id
event_id
instrument_id(object_id)

Event-Beneficiaries
id
event_id
beneficiary_id(object_id)

Event-Locations
id
event_id
location_id(object_id)

Event-Adjectives
id
event_id
object_id
adjective_id

Event-Adverbs
id
event_id
action_id
adverb_id