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 add them to the index store.

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 with time: Show me all the widgets that were selected last Tuesday.

4. Handling complex negation (negation of x AND Y kind of phrases)

5. Intersective vs. Non-intersective adjectives

6. Complex causality handling: The current implementation only allows a primitive causality of ’if then’ kind. Expanding to to more complex causality mapping would allow us handle complex causality.

7. Handling constraints around conjunctions and disjunctions.

The indexer would do a semantic encoding of english text corpus to build its index store. As new data comes in, the index store would get updated. For querying, we would allow a web interface that can take natural language query and would search the index store to find an answer to the query. We would also provide APIs for building the index and a SQL/python interface for searching the index.