2009 IEEE International Conference on Semantic Computing

Query Sentences as Semantic (Sub) Networks

Joel Booth, Barbara Di Eugenio, Isabel F. Cruz, Ouri Wolfson
University of Illinois at Chicago
Chicago, IL, USA
jbooth3@uic.edu, bdieugen@cs.uic.edu, ifc@cs.uic.edu, wolfson@cs.uic.edu

Abstract

This paper presents an approach for representing queries in natural language as semantic networks. The semantic representation is intended to facilitate the translation between natural language and database queries. A domain agnostic algorithm, based on shallow features, is used to map a sentence to a sub-network of concepts within a larger ontology. This work focuses on transportation information systems and gives promising preliminary results.

***Keywords*-natural language processing, query representation, ontologies, natural language interfaces;**

I. Introduction

Natural language provides a powerful and intuitive input modality for users. The expressive power is tremendous and the users are already familiar with the language. Unfortunately, these same features that make natural language so powerful also make its understanding difficult. In this paper, we present a domain agnostic approach of interpreting the explicit and implicit semantics of a natural language query in the presence of a domain ontology. By domain agnostic we mean that the algorithm does not depend on the domain of the ontology being used. The algorithm relies only on the annotation (discussed in Section III-C) and concept labels specified in the ontology.

Because testing the algorithm independent of any domain is impossible, we will introduce it in the domain of transportation – specifically, this preliminary work is intended to model the semantics of a natural language query to a transportation information database system [1]. In the future, these results will be used for the development of a fully functional natural language interface to the transportation database system. Within this domain, we will demonstrate how we can derive detailed semantic representations of queries such as:

- *When will the bus arrive?*
- *Which train station is closer, Clark and Lake or Washington and State?*
- *What is the fastest way home today?*

The primary contributions of this research are 1) a classification of query sentence classes based on shallow linguistic cues 2) a domain agnostic algorithm for mapping query sentences to ontological concepts, and 3) a method for describing query semantics as connected subsets of an ontology. The paper is structured as follows: first we discuss related work, then we introduce our test domain and the proposed semantic alignment algorithm. We then present the initial evaluation of the algorithm before providing concluding remarks and directions of continued research.

II. Related Work

Natural language interfaces (NLI) is a well studied field that has enjoyed a resurgence in interest in recent years thanks to the availability of cheap computational power, ubiquitous mobile devices, and significantly improved voice recognition software. Despite having been studied for several decades [2], [3], natural language interfaces to database systems (NLDB) are still difficult to construct. One of the problems they suffer from is the lack of portability across systems and domains. Another significant problem is the specificity of the grammar (or recognized language) that the systems use. A system that works well for a database containing employee information would have to be redesigned if it were to access a database regarding transit schedules. This results in rigid grammar, constrained vocabulary, and a steep learning curve. It is for this reason that we attempt to remove as many grammatical

This research was supported in part by NSF Award DGE-0549489. Additional support for Isabel F. Cruz provided by NSF Awards ITR IIS-0326284, IIS-0513553, and IIS-0812258. Additional support for Ouri Wolfson provided by NSF IIS awards 0847680, 0513736, 0326284.

978-0-7695-3800-6 2009

DOI 10.1109/ICSC.2009.84

and vocabulary constraints as possible. Rather than rely heavily on the syntactic structure, our approach differs in that we utilize dependency relationships which are not as sensitive to the overall sentence structure.

More recently, the community has seen an increase in NLIs to semantic web systems. NLP-Reduce [4] is a domain-agnostic NLI to semantic web ontologies. It does not rely on complex natural language processing, nor does it use a grammar. It extracts keywords and attempts to generate simple queries. AquaLog [5] and its successor PowerAqua [6] are question answering systems that operate on semantic web ontologies. Like our work, it takes full sentences for the input query and uses WordNet to expand the vocabulary that is understood. However, their approach does not generate a rich semantic representation of the complete query.

There have been many approaches to representing query semantics: first order logic, intensional logic, semantic frames, and Montague semantics. A more detailed study of these techniques can be found in Clifford's text [7]. As the more recent NLI work has moved towards applications on the semantic web, we have chosen to utilize ontologies as a powerful and flexible semantic representation model. Importantly, the ontologies can be used to represent the query itself as well as the greater context in which it exists (both within the system and the real world – i.e., the user).

This contextual information is important because users will often make references that require context to be understood. In linguistics, these terms are known as deixis. These contexts may be time, location, or speaker dependent. Winter [8] showed that being able to interpret this type of language is important in route planning – which is a vital component of our larger research focus.

It is apparent that our algorithm is similar to a number found in the ontology alignment community [9], but there are a number of significant differences between traditional ontology alignment and the problem we face: 1) there is no chance for human intervention for fixing bad matches, 2) there is no a-priori ontological structure on the input side, and 3) the input concepts must be a subset of the target ontology concepts.

Finally, we mention our previous work in the development of a database model and query language targeted for use in transportation information systems. The extended SQL-like language [1] provides a means to describe both the spatial, temporal and probabilistic aspects of trips throughout a transportation network.

III. Algorithm

A. Overview

The purpose of the algorithm is to process a syntactic parse-graph and determine where the concepts[1] within that graph are represented within the semantic model. In order to motivate the algorithm in the remainder of this section, we start with an example of the desired output. If we are given a query sentence, *Is there a cheap place to eat on my way home?*, the resulting semantic representation of that sentence is the semantic network shown in Figure 1. The concepts drawn in dashed line are those that were identified as being in both the semantic model and the initial query sentence. The concept *#PLACE* (in uppercase) was identified as being the target of the query (i.e., what the user is looking for). The remaining concepts, in black line, are implicit in the meaning of the query, but can only be utilized when given the relevant semantic context.

B. Parsing

The initial input to the system is the natural language sentence and the ontology. The sentence is parsed using the Stanford Parser [10], [11]. The output of the parsing is a graph that combines a standard syntactic tree tagged with part-of-speech (POS) labels and edges representing dependency[2] relations. The remainder of this section provides the details of how this syntactic parse, in conjunction with the semantic model, is mapped into a rich semantic representation.

C. Domain Model

The algorithm itself is domain agnostic, but for the purpose of evaluation we have chosen to develop an ontology for the transportation domain. This ontology contains concepts related to the physical network (e.g., links, intersections, speeds), resources of interest (e.g., grocery stores, banks), events (e.g., starting, stopping, transferring, shopping, working), and route-based concepts (e.g., trips and their sub-components). In addition to the transportation concepts we model a number of what could be considered linguistic deixis – terms like prepositions (e.g., along, near, around) and qualitatives (e.g., best) that can be used in conjunction with concepts within the transportation system. The final concept is that of a user, which has certain properties and is assumed to interact

[1]For the remainder of the paper, concepts will include both classes and properties.

[2]The dependency relations represent how the words are related regardless of the syntactic structure. Examples include direct object and modifying relationships.

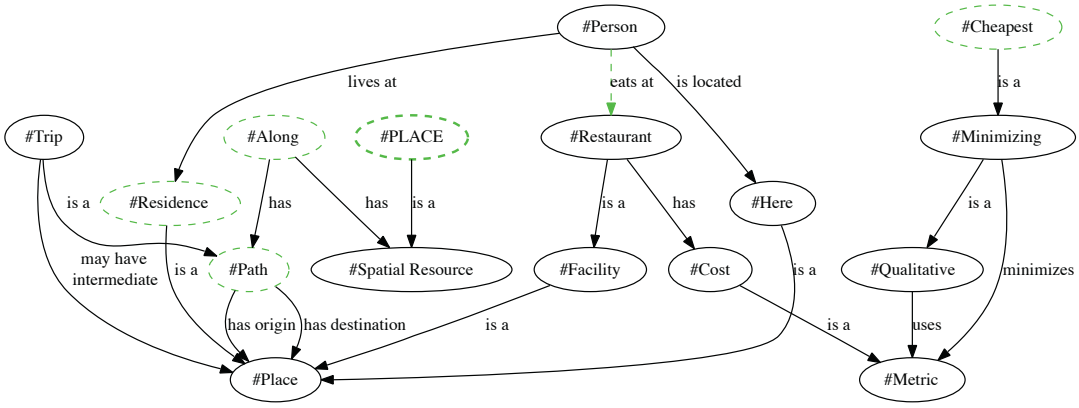

Figure 1. Semantic Network Representation

with the system. Broadly speaking, these concepts can be divided into categories of abstract, spatial, temporal, and spatio-temporal.

The concepts and properties within the ontology were manually annotated with their corresponding WordNet [12] entries wherever a suitable entry exists. Additionally, some of the classes and properties have been annotated with specific colloquialisms or non-literal synonyms where appropriate. This is done because no linguistic resource maintains a list of local colloquial language. The annotation is used in the alignment algorithm as a way of extending the vocabulary used as well as providing some level of semantic information.

D. Candidate Set Generation

After the initial syntactic parse, the next task is the generation of the *candidate set* – the set of concepts within the ontology that may align to terms in the query sentence. This is done through a pair-wise comparison between a subset of the sentence graph nodes and the concepts in the ontology. Noun, compound noun, conjunctive noun phrase, verb, adjective, adverb, and preposition nodes are included in the search.

A list of *stop words* are ignored in the initial input. The stop words used are not the same as those commonly found in other natural language processing tasks as the generic stop word lists include prepositions and other words that carry important semantic information. The current list of stop words for our algorithm is *[are, is, does, the, a, there, what, where, when, which, why, me, my, find, locate, get, tell]*. These are common words that do not align with any specific concept in the ontology.

In generating the candidate set there are two primary comparison methods: plain text-based and WordNet-based. For the plain text comparison the surface form of the word (i.e., the word itself) is compared against the label for the concept in the ontology. The comparison is done by a direct equivalence match, edit distance match, and stemming match. The first simply checks whether the two strings are identical (case insensitive), the edit distance checks for equivalence within an edit distance of k, and the stemmed match checks whether the label stems[3] match. The same string metrics are applied to the colloquial annotation and the node surface text. Similarly, nodes are matched against the *instances* in the ontology. For example, *Sears Tower* would align to the name of a specific building (i.e., an instance of a concept).

The second type of matching relies on WordNet to determine synonymy. As stated in Section III-C, most of the concepts in the ontology were labeled with an appropriate WordNet index and sense number. The index is a pointer to a set of synonyms for the word and is indexed by a word (lemma) and part-of-speech. For example, in our ontology the concept *bus* has the index *[bus, noun]* and sense number 1. This points to a synset with the following terms {*bus, autobus, coach, charabanc, double-decker, jitney, motorbus, motor coach, omnibus, passenger vehicle*}.

For the nodes in the parse graph we generate WordNet information in two ways. The first, and likely more accurate, method uses the surface form of the word and the part-of-speech tag returned by the parser to lookup the appropriate WordNet index. The second method uses only

[3]Stemming is a common technique that attempts to extract the root of the word. For example, *matching, matcher,* and *matches* all have the stem *match*.

the surface word and no part-of-speech information and returns a set of WordNet indices. It should be noted that for either method we lack knowledge of the appropriate sense number for the indices. That means that *bus* may refer to a computer bus (i.e., a device that transfers data between components) as well as the vehicle. We ignore this problem for now as we are generating *candidates* for matches.

For each of the potential indices of the node, the synonym, hypernym, and hyponym synsets are enumerated. If any of these matches the pre-defined synset for the ontology concept it is considered a candidate.

E. Candidate Selection

At the end of the candidate generation phase there will likely be multiple candidates for some nodes, and some or all of these candidates may be incorrect. For that reason we need a ranking and selection metric to determine which candidates are true matches for any given node. We have defined the list of match types, and through empirical tuning have set their corresponding weight, that can be found in Tables I and II.

Concept Target String	Distance Measure	Weight
Plain Text$_{ConceptLabel}$	Equal	1
Plain Text$_{ColloquialLabel}$	Equal	1
Plain Text$_{InstanceLabel}$	Equal	1
Plain Text$_{ConceptLabel}$	Stemmed Equal	.25
Plain Text$_{ColloquialLabel}$	Stemmed Equal	.25
Plain Text$_{InstanceLabel}$	Stemmed Equal	.25
Plain Text$_{ConceptLabel}$	Edit Distance: 1	.15
Plain Text$_{ColloquialLabel}$	Edit Distance: 1	.15
Plain Text$_{InstanceLabel}$	Edit Distance: 1	.15
Plain Text$_{ConceptLabel}$	Edit Distance: 2	.05
Plain Text$_{ColloquialLabel}$	Edit Distance: 2	.05
Plain Text$_{InstanceLabel}$	Edit Distance: 2	.05

Table I. Weights for String Matches

Node Source	Concept Target	Weight
Synonym$_{Lemma+POS}$	WNI	.5
Hypernym$_{Lemma+POS}$	WNI	.1
Hyponym$_{Lemma+POS}$	WNI	.1
Synonym$_{Lemma}$	WNI	.25
Hypernym$_{Lemma}$	WNI	.05
Hyponym$_{Lemma}$	WNI	.05

Table II. Weights for WordNet Matches

For each candidate, the match weights are summed to give the cumulative match score. The concept that has the maximum cumulative weight is considered the correct match for the node. There is a minimum threshold of .24 for any match to occur.

F. Query Target Identification

Now that the sentence has been mapped to a set of concepts within the ontology, we must identify the target concept for the query. In our ongoing example the *place* is the query target as it is the concept for which the user is searching. Through empirical testing, we found that there are several classes of queries for which the target can be identified using shallow linguistic features. For each class of query targets we have determined a set of identifying cues. The cues present in any input query determine which method is used for selecting the query target.

1) Copula: The first class of queries can be identified by their overall structure. Queries of the form *What is/are ...?* and *Is/Are there ... ?*, such as

- *What is the fastest way home today?*
- *Is there a grocery store in this neighborhood?*

are handled by identifying the copular[4] dependency of the conjugated verb *to be*. In the case of our ongoing example (*Is there a cheap place to eat on my way home?*) this is indicated by an edge labeled *cop* between *is* and *place* in the dependency graph produced by the syntactic parser.

2) Nominal Subject: Similar to the class of queries that can be identified by a copular dependency, another class can be identified though nominal subject[5] dependencies. The most important construct in this class is *Where is/are... ?*, such as the queries:

- *Where is the cheapest gas station?*
- *Where are nearby parks?*

The query targets in these sentences can be identified by looking at the nominal subject dependency of the verb *to be*. In the first example the dependency is *is→station*.

3) Copula + Nominal Subject: Queries of the form *Which X is ... ?"*, where *X* is some arbitrary resource, have slightly more complicated structures than those descussed thus far, but we can still identify the query target using the dependency parse. In queries like,

- *Which train station is closer, Clark and Lake or Washington and State?*
- *Which street near my house is the least congested?*

we can identify the query target by using two dependencies: copula and nominal subject. In the first example, the copular dependency is *is→closer*; similarly, the dependency in the second example is *is→congested*. In both cases the target of the copular dependency is in a second dependency relationship of nominal subject. In the first example we see the nominal subject dependency *closer→station* and in the second example *congested→street*. This dependency chain generalizes and we

[4]A copula is a word used to connect a subject with a predicate – even if there is no action or condition between the subject and predicate. A copular verb is often called a *linking verb*.

[5]A nominal subject is a noun phrase that is the subject of a clause.

are able to identify the query target finding the nominal subject dependency of the copular dependency of *to be*.

4) Direct Object: Another large class of queries is characterized by the type of verb used. *Find, locate, determine, get, give* and others all explicitly specify a direct object of interest. Once again, this target can be identified using the dependency parse – the direct object of the head verb points to the query target. Examples of such queries include:

- *Find a pharmacy near my house.*
- *Give me the closest ATM.*

In the first example the direct object dependency is as follows: *find→pharmacy*.

5) Temporal: The final class of query targets we can identify are those that have a temporal query target. Examples include,

- *What time does my train depart?*
- *When will the bus arrive?*
- *How long is the wait?*

These queries are not covered by the classes above, but they do possess a *temporal* concept from the ontology. If all other cases fail, and there is a temporal concept present, we assume that the temporal concept is the target of the query. Temporal concepts can be identified as they are a subclass of the *Temporal* concept in the ontology.

G. Result Generation

The final phase of the algorithm is the generation of the semantic sub-network. We have identified both the set of ontology concepts present in the query as well as which concept is the target of the query. Given this information, we can find the concepts and properties that link them together.

The set of concepts exist within the ontology, and we execute a frontier search to find paths through the ontology that connect the concepts. We perform a breadth-first expansion (treating the directed ontology network as an undirected network) to a depth of three for each class. Similarly, for properties we perform the same expansion starting with the classes that the property relates. This process produces a small, local network for each concept. We take the union of the local networks and then prune all nodes that do not lie on a path between the concepts identified in the query sentence.

IV. Results

In this section we will discuss the match weight tuning and overall system performance. For testing and tuning the algorithm we focused on the twelve example sentences presented throughout this paper. Each sentence was successfully parsed and represented as a parse-graph. After pruning for stop words, a total of 43 nodes were identified for the candidate generation phase. Candidates for each node were generated using the algorithm described in Section III-D. More than half of the nodes (25/43) had multiple candidate matches. More details can be found in Table III.

Nodes Tested	43
Mean Candidates per Node	2.76
Median Candidates per Node	2
Std. Dev. Candidates per Node	2.61
Max Candidates per Node	11
Min Candidates per Node	1
Nodes with 1 Candidate	18

Table III. Candidate Generation Statistics

Once the candidate generation phase was completed we manually analyzed the data in order to determine the appropriate match weights. A "gold standard" alignment was generated by hand so that there would be a metric to score the resolution algorithm. Weights were changed manually over multiple iterations until the best performance was achieved.

Nodes Correctly Resolved	39
Nodes Semi-Correctly Resolved	3
Nodes Incorrectly Resolved	0
Nodes Correctly Unresolved	1
Nodes Incorrectly Unresolved	0
Total Nodes	43

Table IV. Candidate Resolution Statistics

Shown in Table IV is the performance of the candidate resolution algorithm. On this set of input the algorithm is overwhelming successful with an accuracy of 93%. There were no completely wrong resolutions. The three semi-correct resolutions are the result of three nodes being mapped to a concept that was too general. In one case a node should have been mapped to a *station* but was mapped to a *facility* instead. The mapping is semantically correct, but the more specific concept would be preferable. The correctly unresolved node was one that had generated candidates even though there was no appropriate concept in the ontology. The weight of the match was not greater than the threshold needed so it was not resolved to any concept.

The next phase of the algorithm was the detection of the query target. For these test sentences, all of the query targets were successfully identified. As with the other phases of the algorithm, the current approach appears to work well for connecting the concepts. It may be the case that for certain inputs the connections include too many or too few concepts.

The results thus far show that this is a promising approach to generating an ontology-based representation for single-sentence natural language queries. The test sentences used were selected such that their concepts are present in the ontology – this did not necessarily mean that those concepts would be identified correctly. It will always be the case that concepts beyond the scope of the ontology cannot be mapped correctly.

V. Conclusions and Future Work

We have presented a domain agnostic algorithm for representing single sentence queries as semantic networks based on a domain ontology. This makes the implicit sentence semantics explicit in the final representation. The ontology provides semantics for generic spatio-temporal relationships as they apply to language. Knowing the semantics for a word such as *along* means that the system can look for, and use, the appropriate concepts elsewhere in the sentence.

We have achieved interesting and promising results thus far, but we have a number of directions that we would like to explore in the future. At this time we do not make use the WordNet gloss information, nor do we use additional resources like SUMO [13]. These would allow for richer candidate set generation. Additionally, the candidate generation phase could be improved with better string metrics, more advanced WordNet comparisons, and other techniques from the ontology alignment community. The current candidate selection algorithm relies only on weights attributed to match types. Focus should be put on minimizing the distance between concepts within the ontology. Intuitively, the concepts in a given query should be semantically close, and therefore they should be close within the ontology.

We will also continue extending the coverage of the ontology (as well as trying different domain ontologies) and explore additional grammatical cases that can be parsed. We also understand that the current level of evaluation is insufficient to draw any concrete conclusions beyond the fact that this approach looks promising. A larger corpus of queries is required for further evaluation. The most important class of language in need of exploration is that of the more complicated trip queries mentioned in the related work. We would also like to look at what happens when the input consists of multiple sentences or sentence fragments.

As stated in the introduction, the ultimate purpose of this semantic representation is the facilitation of generating database queries from natural language input. The most interesting, and demanding, future work will be leveraging our representation in building a functional NLDB system for transportation.

References

[1] J. Booth, P. Sistla, O. Wolfson, and I. F. Cruz, "A data model for trip planning in multimodal transportation systems," in *12th International Conference on Extending Database Technology (EDBT)*, Saint Petersburg, Russia, 2009, pp. 994–1005.

[2] A.-M. Popescu, O. Etzioni, and H. Kautz, "Towards a theory of natural language interfaces to databases," in *8th International Conference on Intelligent User Interfaces*, New York, NY, USA, 2003, pp. 149–157.

[3] S. J. Kaplan, "Designing a portable natural language database query system," *ACM Transactions on Database Systems*, vol. 9, no. 1, pp. 1–19, 1984.

[4] E. Kaufmann, A. Bernstein, and L. Fischer, "NLP-Reduce: A "naïve" but domain-independent natural language interface for querying ontologies," in *4th European Semantic Web Conference (ESWC)*, Innsbruck, Austria, June 2007.

[5] V. L. Garcia, E. Motta, and V. Uren, "AquaLog: An ontology-driven question answering system to interface the semantic web," in *North American Chapter of the Association for Computational Linguistics Conference on Human Language Technology (HLT-NAACL)*, New York, NY, USA, 2006, pp. 269–272.

[6] V. Lopez, E. Motta, and V. Uren, "PowerAqua: Fishing the semantic web," in *3rd European Semantic Web Conference (ESWC)*, Budva, Montenegro, June 2006, pp. 393–410.

[7] J. Clifford, *Formal Semantics and Pragmatics for Natural Language Querying*, ser. Cambridge Tracts in Theoretical Computer Science. Cambridge, MA, USA: Cambridge University Press, 1990, no. 8.

[8] S. Winter, "Towards a conceptual model of talking to a route planner," in *8th International Symposium on Web and Wireless Geographical Information Systems (W2GIS)*, Shanghai, China, December 2008, pp. 107–123.

[9] P. Shvaiko and J. Euzenat, "A Survey of Schema-Based Matching Approaches." in *Journal on Data Semantics IV*, ser. Lecture Notes in Computer Science, vol. 3730. Springer, 2005, pp. 146–171.

[10] D. Klein and C. D. Manning, "Fast exact inference with a factored model for natural language parsing," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 15. MIT Press, 2002, pp. 3–10.

[11] ——, "Accurate unlexicalized parsing," in *41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, July 2003, pp. 423–430.

[12] Fellbaum, *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.

[13] I. Niles and A. Pease, "Toward a standard upper ontology." in *2nd International Conference on Formal Ontology in Information Systems (FOIS)*, Ogunquit, Maine, USA, October 2001.