Abstracting Natural Language Queries into SQL

Thomas Hagen

Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
hagen715@morris.umn.edu

November 12th, 2016

Big Picture

We're talking about a Natural Language Interface that:

- Has access to a structured database.
- Takes in a question in English from a user.
- Tries to interpret the question.
- Returns information from the database as an answer.
- Conceptually similar to Siri/Alexa.

Outline

Plan

What is SQL?

Before we get into some code examples:

- Structured Query Language (SQL) is a special-purpose programming language designed for managing data held in a relation database management system (RDBMS)
- A relational database can be thought of as a collection of spreadsheet tables containing rows, each row with a unique key, and columns.
- SQL gives outside entities a way to add, modify, initialize, and query databases.

SQL Code

```
SELECT *
 FROM  Books;
```

- SELECT: every (*) column
- FROM: the Books table

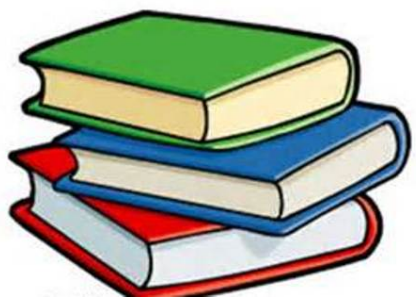
→

SQL Code

```
SELECT *
 FROM  Books
 WHERE price > 100;
```

- SELECT: every (*) column
- FROM: the Books table
- WHERE: the book's price is over 100.

→



SQL Code

```
SELECT *
 FROM  Books
 WHERE price > 100
 ORDER BY title;
```

- SELECT: every (*) column
- FROM: the Books table
- WHERE: the book's price is over 100.
- ORDER BY: title, alphabetically

→

What is Natural Language?

- A natural language or ordinary language is any language that has evolved naturally in humans through use and repetition without conscious planning or premeditation.
- In contrast to formal or constructed language.
- Converting from natural to formal requires language analysis.

Morphological Analysis

- Used to represent the meaning and grammatical features of the word.
- Splits words into their prefixes, roots, and suffixes.
- "Independently" can be deconstructed into to In-depend-ent-ly
- Words can be identified easily and mapped against existing bodies of knowledge.

Lexical Analysis

- A lexicon is the vocabulary of a person, language, or branch of knowledge.
- "Properties" such as its type (noun, verb, adjective), synonyms, antonyms, and homonyms.
- Universal lexicon is general, domain lexicon is subject-specific.

Syntactic Analysis

- Determine the intent or specific meanings of the individual words in context to the query as a whole.
- Probabilistic Context-Free-Grammars (PCFGs) use large sets of rules and probabilities to guess the most likely sentence structure.

Plan

Unrestricted Approach

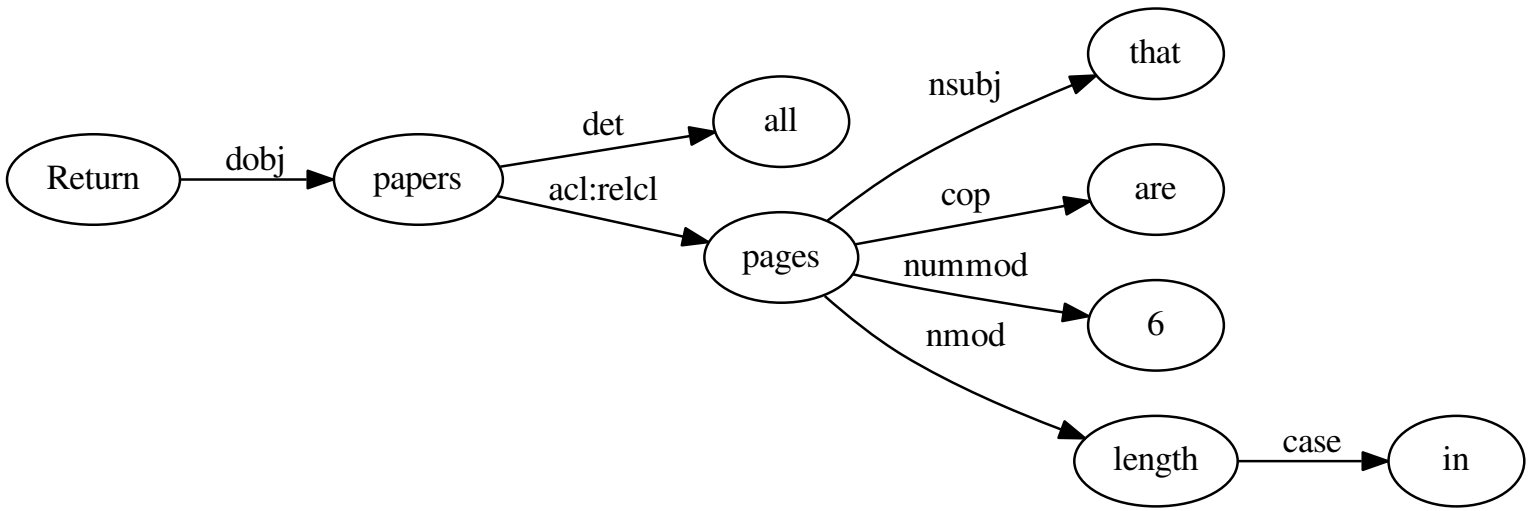
Natural Language Interface to Relational database: NaLIR

- Developed at University of Michigan
- Has active user feedback
- Unrestricted language input

Word Relationships

- Start by decomposing sentences using language analysis.
- The core of sentence decomposition lies in how we represent sub-relationships between individual word pairs.
- The Stanford Parser uses a fixed collection of 44 hierarchical relationships that define how one word relates to another.

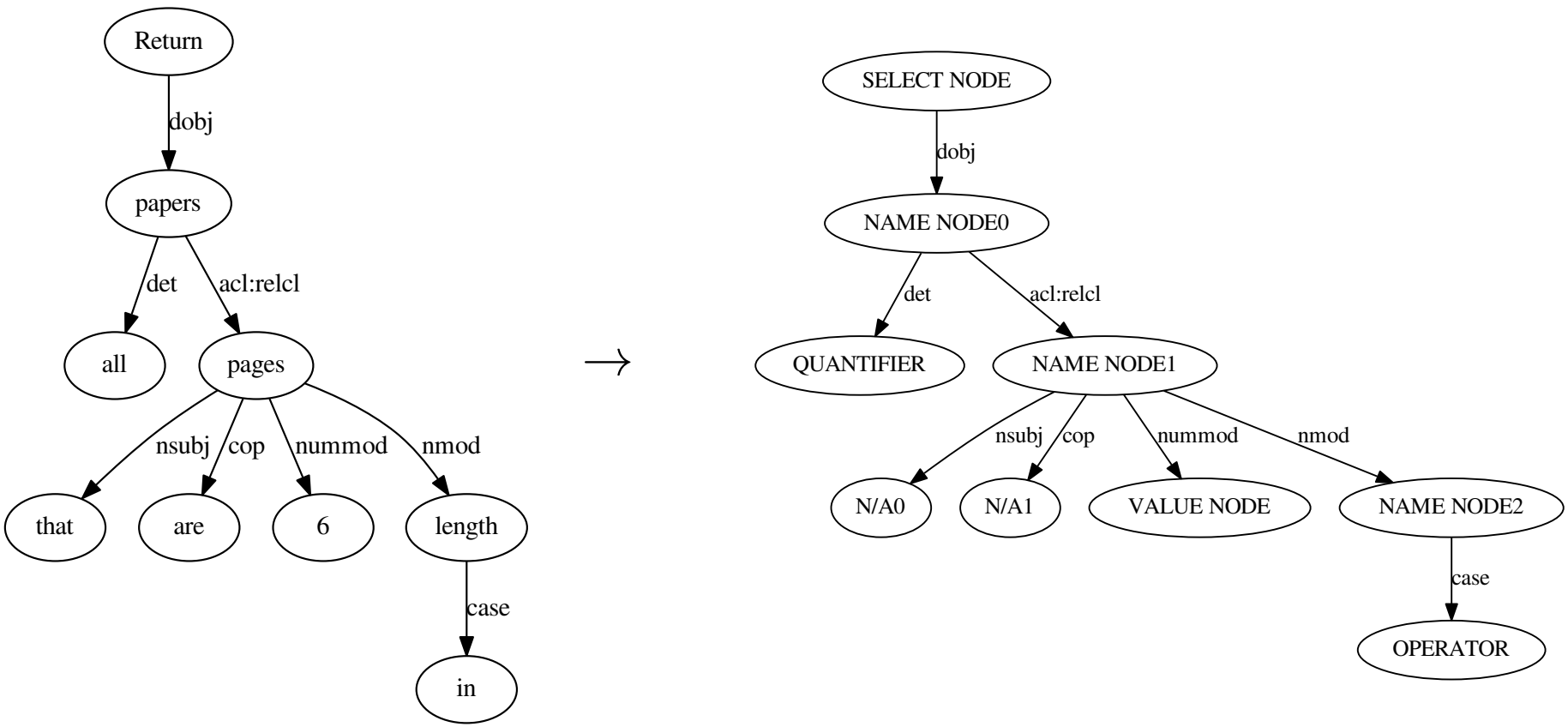
Dependency Tree



- "Return all papers that are 6 pages in length."
- Also called a dependency-based parse tree.
- "Papers" is the direct object of "Return", and in the tree it is the direct child by the dobj edge.

What is a Query Tree?

- Intermediate representation between parse tree and SQL
- Map the dependency tree nodes to quantifying nodes.



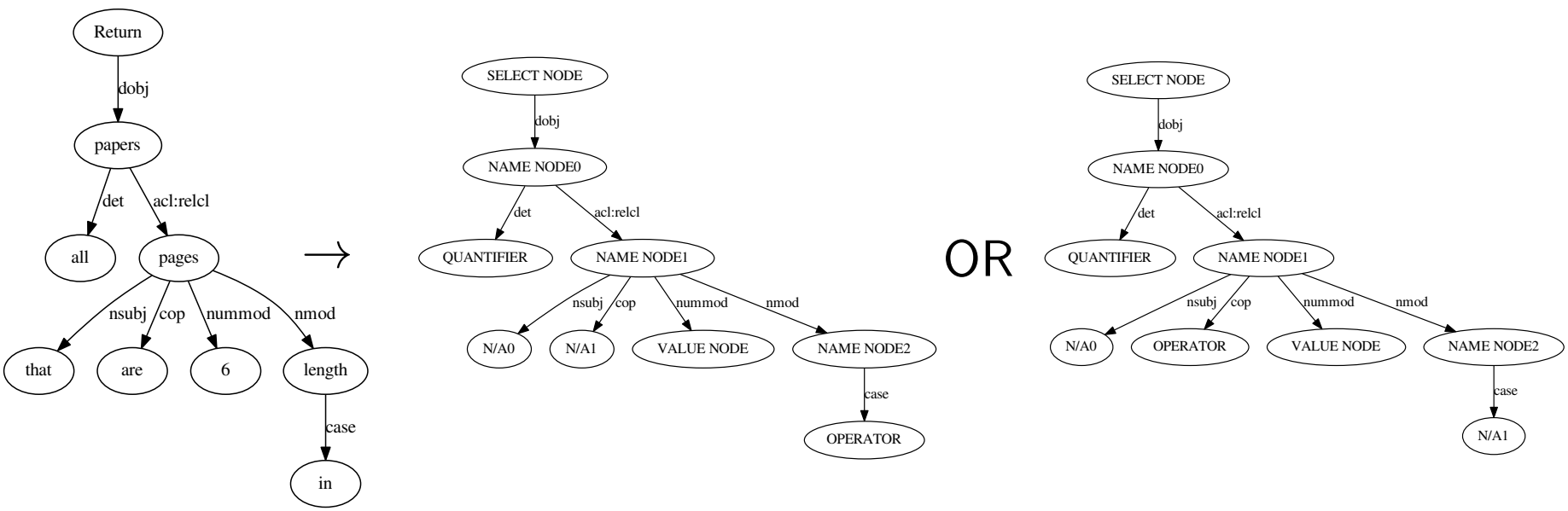
Quantifying Nodes

Node Type	**Corresponding SQL**
Select Node	SQL Keyword: SELECT
Operator Node	an operator, eg. $=$, $>=$, $!=$
Function Node	an aggregation function eg. AVG
Name Node	a table name or column name
Value Node	a value under a column
Quantifier Node	ALL, ANY, EACH
Logic Node	AND, OR, NOT

- How does NaLIR classify nodes?

Quantifying Node Class

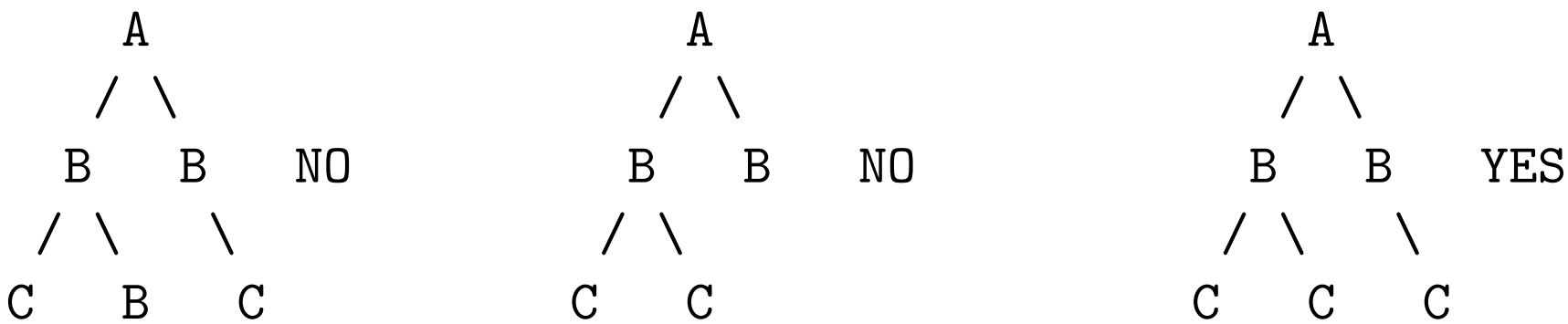
- Word similarity is evaluated using a universal lexicon.
- Jaccard Coefficent is used to evaluate spelling similarity.
- Nodes might qualify to be in multiple classes.



Choosing Query Trees

- Query trees can match English structure but not SQL
- Valid SQL structure for NaLIR is defined by a simple grammar
- Set of valid trees returned to the user to choose from.

```
Ex: 0. Must start with A
    1. A -> B
    2. B -> A | C
    3. C -> Leaf Node
```



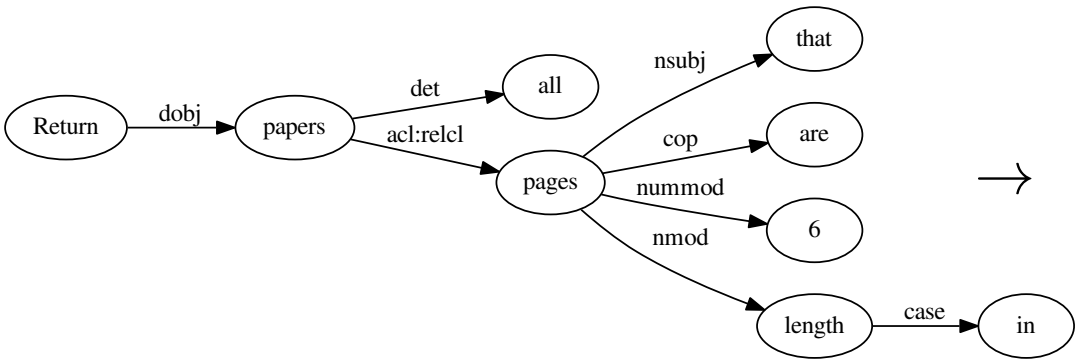
Query Tree to SQL

The final query tree is parse through to generate SQL

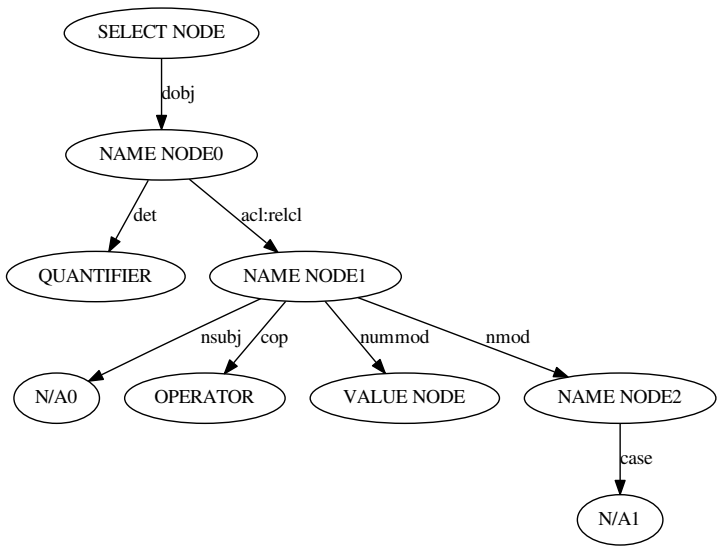
- SELECT node is identified
- Operation Nodes and Value Nodes are added to WHERE.
- Name nodes are identified as columns and tables.
- In complex queries, sub-queries from Function Nodes are evaluated and stored using AS.

Natural Language Query:
‘‘Return all papers that are 6 pages in length."

Dependency Tree:



Query Tree:



SQL:
```
SELECT *
  FROM Papers
  WHERE length = ‘6’;
```

Plan

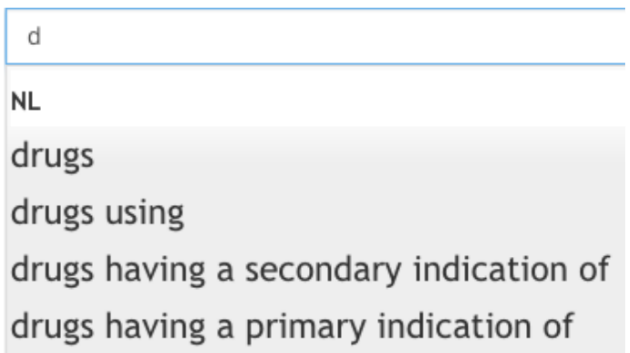
Auto-Suggestion Queries

TR:Discover

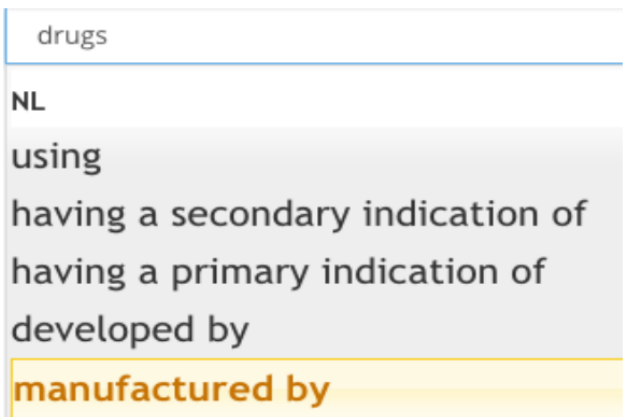
- Developed by Thomson Reuters
- Focus on pharmaceutical patents
- Mix of Keyword and Unrestricted approaches

Auto-suggestion Example

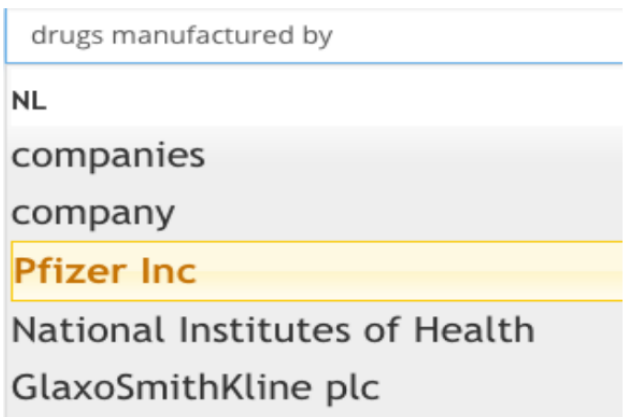
1.



1. Auto-suggestions for “d”
2. Phrases that follow “drugs”
3. The grammar dictates a company name must follow



2.



3.

*Taken from [12]

Domain Lexicon Benefits

- *Discover* uses a domain lexicon, meaning it is specific to the database materials
- Lexicon contains pre-defined suggestion segments which may contain more than one word, such as “developed by”.
- Grammar rules are a mix of universal English rules as well as rules specific to the database.
- Ex. What suggestion segments can follow or precede the segment “headquartered”.

First-Order Logic (FOL) Parser

- Generate a syntax tree using a generic parser such as ANTLR, the one utilized by *Discover*.
- Takes in the FOL representation and the grammar.
- Grammar is comprised of the set of all rules used to generate the query and the lexicon used.
- With these pieces of information, the parser will attempt to determine a parse tree.

FOL Tree to SQL

- By limiting the grammar, the parse tree is already correct.
- From a parse tree, conversion to SQL is similar to NaLIR.
- Elements are mapped to their corresponding attributes in the database.

```
Natural Language Query: drugs developed by Merck

FOL: all x.(drug(x) ->
     (develop_org_drug(id0,x) & type(id0,Company)
     & label(id0,Merck)))

SQL: SELECT drug.*
       FROM drug
       WHERE drug.originator-company-name = 'Merck'
```

Plan

Conclusion

- Universal vs Domain lexicons
- Useful technology in a limited field
- Future of Structured Data Query
- Go access a database today!

Questions?

Thank you to Nic Mcphee and KK for constructive feedback, as well as alumni. Thank you to my parents for driving three hours.

Questions?

References I

[1] Constructing an interactive natural language interface for relational databases Li, Fei, and H. V. Jagadish. "Constructing an interactive natural language interface for relational databases." Proceedings of the VLDB Endowment 8.1 (2014): 73-84.
[2] A Fast and Accurate Dependency Parser using Neural Networks Chen, Danqi, and Christopher D. Manning. "A Fast and Accurate Dependency Parser using Neural Networks." EMNLP. 2014.
[3] Generating Typed Dependency Parses from Phrase Structure Parses De Marneffe, Marie-Catherine, Bill MacCartney, and Christopher D. Manning. "Generating typed dependency parses from phrase structure parses."Proceedings of LREC. Vol. 6. No. 2006. 2006.

References II

[5] Natural language and keyword based interface to database Shah, Axita, et al. "NLKBIDB-Natural language and keyword based interface to database." Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on. IEEE, 2013.
[6] A WordNet-based natural language interface to relational databases Hu Li and Yong Shi, "A WordNet-based natural language interface to relational databases," Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on, Singapore, 2010, pp. 514-518.
[7] Z. Wu and M. S. Palmer. Verb semantics and lexical selection. In ACL, pages 133–138, 1994.

References III

[8] C. Xiao, W. Wang, X. Lin, J. X. Yu, and G. Wang. Efficient similarity joins for near-duplicate detection. ACM Trans. Database Syst., 36(3):15, 2011.
[9] "SQL" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc., date last updated (9 10 2016). Web. Date accessed (10 October 2016). ¡https://en.wikipedia.org/wiki/SQL¿
[10] Natural language Interface for Database: A Brief review Nihalani, Mrs Neelu, Sanjay Silakari, and Mahesh Motwani. "Natural language interface for database: a brief review." (2011).
[11] Li, Xian, Weiyi Meng, and Xiaofeng Meng. "EasyQuerier: a keyword based interface for web database integration system." International Conference on Database Systems for Advanced Applications. Springer Berlin Heidelberg, 2007.

References IV

[12] Song, Dezhao, et al. "TR Discover: A natural language question answering system for interlinked datasets." The 14th International Semantic Web Conference. 2015.
[13] Barwise, Jon. "An introduction to first-order logic." Studies in Logic and the Foundations of Mathematics 90 (1977): 5-46.
[14] Princeton University "About WordNet." WordNet. Princeton University. 2010. ¡http://wordnet.princeton.edu¿