Team: **Humean Hackers**
Silvia Colombo
Vanessa Hanschke

Knowledge Extraction

Data and Information Integration
Deliverable 1

Introduction

The aim of the project is to create a software that analyses natural language text and extracts relevant information in a structured form that can be further analysed by a machine.

Methods and Tools

The program uses NLTK (Natural Language Toolkit), a natural language processing tool that uses Python. The output should be stored in a Neo4j relational database. This graph database consists of nodes representing entities such as places or people and actions which will be named using the predicates of a sentence. A group of two nodes connected by a relationship represents a fact. The number of times a relationship between two nodes is visited, will be counted and used to statistically judge the validity of the fact.
The Database that has been chosen to do the experiments on is a Yahoo! Labs Database. It is comprised of 142,627 questions and their answers, thus offering a large database of knowledge to extract. The selected questions and answers have been chosen for their linguistic properties, only including at least four word sentences with a noun and a verb. Furthermore, some metadata on categories, ratings etc. is included

Execution

The main focus in this stage of the project has been processing plain language text and extracting as much information as possible by identifying entities and the relations between them. For this, the text is read from a file and is stripped of irrelevant data and a file is saved with only the relevant information one sentence for every line, for example metadata. Since the dataset is expected to be quite large, this will be done to save time while running the code. Next, the text is split into sentences, the sentences are split into tokens, which in the end will be tagged according to their Parts-of-speech. This will constitute the needed input for the main function of the software: entity detection and relation detection. The method used to define entities has been taken from “Natural Language Processing with Python – Analysing Text with the Natural Language Toolkit” and is called ‘Chunking’. Here we group the different tokens into noun phrases and prepositional phrases, which can then be assigned to nodes and verb phrases which are used to assign actions.

Unfortunately, with the authorisation to use the data coming too late from Yahoo and, especially due to the difficulties we have encountered setting up the Neo4j database, the writing of the code had been held back.

Team: **Humean Hackers**
Silvia Colombo
Vanessa Hanschke

Discussion

The modeling and planning stage brought many important questions to the table. The biggest difficulty is to get around the peculiarities of human language, especially those on an informal platform such as a question-answer website, or the world wide web in general.
One big difficulty are pronouns and contexts, i.e. pragmatics. Many times the subjects of sentences are inferred or further specified by the sentences that precede them.

Example 1: Obama is the president of the United States. He lives in the White House.

We would like to extract the fact that 'Obama lives in the White House'. Hence, it would be desirable to not see "He" as a separate node but as referring to "Obama". This could maybe done through Named Entities, which could refer personal pronouns to chunks that precede them which include persons as Named Entities.

Sometimes context also changes the meaning of the words that are used.

Example 2: Penguins mate in the winter. The male takes care of the egg.

Extracting the information "Male - takes care of -> Egg" would therefore be a false assumption as this fact only holds in the context of penguins. This is a pragmatic problem which is harder to solve and might have to be overlooked.

Another problem is the meaning of the adverb 'not', which is tagged as a normal adverb, but changes the meaning of a sentence completely.

Moreover, a way of joining the nodes which are the same semantically, but differently syntactically, has to be found. If one takes the example 'President Obama' and 'Obama', we see that these are different in form but refer to the same real world entity. Also verbs with tense marking will not be joined if not spelled exactly the same. Even spelling mistakes can create several different entities that could all be summed up in one node. To abolish grammatical errors we have decided to lemmatize verbs after having created a list of actions. Then we plan to merge nodes based on similarity. Concretely, the structures of the names of entities and the relationships will be compared.

Conclusion

Concrete results of the application of the planned methods on the Yahoo! Answers Manner Questions 2.0 Dataset will be needed to continue with the project. Finalizations of the code will decide which methods will be useful to overcome the difficulties of natural human language and how much they impede the extraction of knowledge.

Team: **Humean Hackers**
Silvia Colombo
Vanessa Hanschke

Resources

NLTK 3.0 Documentation
http://nltk.org

Natural Language Processing with Python – Analysing Text with the Natural Language Toolkit
Steven Bird, Ewan Klein, and Edward Loper
http://nltk.org/book/

Embedded Neo4j Python Bindings Documentation v1.9-SNAPSHOT
http://docs.neo4j.org/drivers/python-embedded/snapshot/

Yahoo! Answers Manner Questions 2.0 Dataset
http://webscope.sandbox.yahoo.com/catalog.php?datatype=l