Truthfulness Verification System

Tathagata Dasgupta, ABM Musa
Email: {tdasgu2, amusa2}@uic.edu

1 Introduction

2 Current System

We are currently working to get the T-verifier code running on our systems, with inputs from Xian Li. So the descriptions below lacks some required details and makes some assumptions.

As of now T-Verifier produces 5 alternative statements once it is provided with a doubtful statement (DS), a doubtful unit (DU), data type of the unit etc in the first phase. Though these sentences produced are quite accurate, the facility to pinpoint the correct alternative does exist yet. The project goal, may be formulated as:

"Extract information from Wikipedia and ontologies (such as Yago, DBpedia and Freebase) which helps to verify truthfulness of statements. Describe the extraction algorithm and how the information can be utilized to assist statement truthfulness verification."

3 Propsed System

3.1 System Overview

3.2 Description of extraction algorithm

For disambiguation among the alternative statements, Wikipedia is generally used as an authoritative source. However, the contents of Wikipedia is not available in the form that is consumable in a programmatic format. To address such difficulties a number of projects Yago, DBpedia, Freebase have organized the massive amount of data in a searchable fashion e.g DBpedia uses SPARQL endpoint, Freebase uses MQL api. Open source implementation of Python wrappers exist for both the interfaces exist and appear to be mature enough for our needs.

3.3 Freebase

Freebase has information about approximately 20 million Topics, each one having a unique Id, which can help distinguish multiple entities which have similar names, such as Henry Ford the industrialist vs Henry Ford the footballer. Most of the topics are associated with one or more types (such as people, places, books, films, etc) and may have additional properties like "date of birth" for a person or latitude and longitude for a location.

Python library: http://code.google.com/p/freebase-python/ The type system: http://blog.freebase.com/2007/03/1
type-system-in-freebase/

Freebase seems to be the first point to start not only because of the richness of the data available but also the available api.

3.4 Dbpedia

http://wiki.dbpedia.org/Datasets?v=lbf http://wiki.freebase.com/wiki/DBPedia http://sourceforge.net/projects/
wrapper/files/sparql-wrapper-python/1.4.2/

A few major tasks would be:

3.5 Yago

YAGO is a semantic knowledge base with over 900,000 entities (like persons, organizations, cities, etc.). It knows about 6 million facts about these entities

4 Truthfulness verification

4.1 Building queries from the data supplied by the user

We take the content words from the topic unit(TU), along with each of the alternative units(AU), and plugin the datatype(t) to form a number of queries.

$$q_i = \{TU, AU_i, t_{AU_i}\} \qquad \text{where } i = 1 \cdots n$$

where n is the number of alternative units generated by the first phase of the existing system. By quering the various ontologies described above we will get results which might have the following cases:

$$\begin{aligned} freebase(q_i) &= \phi \\ &= r_i \end{aligned}$$

From what we have explored till now, by quering Freebase we can get the various property, value pairs of the entity that we have supplied in the form of TU. In case the result is ϕ, it suggests that we do not have information for this particulat combination TU and AU_i. Due to the richness of the information in Freebase we can assume with a considerable degree of confidence that this combination is untrue. This is one way verification, but in case of non-null results, how do we filter and rank the results with the available set of information is something we can not formulate at this stage with out further exploration.

4.2 Application of Machine learning algorithms

: Though application of machine learning algorithms like using Naive Bayesian, neural network

5 2.1 Supervised - Use a naive bayesian classifier

6 Unsupervised methods - clustering or Probabilistic LSI

7 Partially Supervised methods