Analogy Engines for the Semantic Web

Akshay U. Bhat [1, 2]
1 Department of chemical engineering, Institute of Chemical Technology, Mumbai, India
2 Syracuse University, Syracuse, NY 13244 USA
001-315-460-0854
Akshayubhat@gmail.com

ABSTRACT

We propose a new utility for Semantic Web called as Analogy Engine. Analogy engine employs an example based search approach for retrieving the most similar URIs for the given URI by comparing number of shared links. The Analogy engine is based on Analogy Space, which uses Singular Value Decomposition on matrix representation of a Semantic Network. However Analogy Space faces difficulty with networks having more than a few thousand nodes. We present our preliminary work on scaling Analogy Space by dividing the network into multiple communities, and creating separate Analogy Space for each community. We show that this procedure results in significant improvements and can be used for a large scale network such as the Semantic Web.

Categories and Subject Descriptors:

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search

General Terms: Algorithms

Keywords: Search Engines, Analogy Space, Semantic Web

1. INTRODUCTION

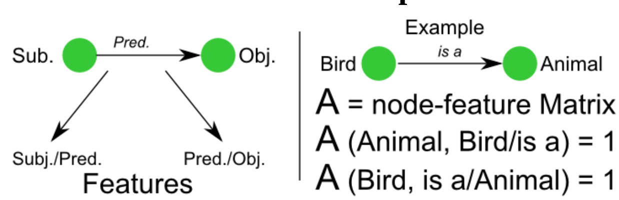
A number of search engines are currently being developed for Semantic Web, most being keyword based search engines. However In this poster we present a new approach for searching/exploring information on Semantic Web. We propose an example based approach which utilizes the network structure to find the most similar URIs for a given URI by comparing number of shared links. E.g. consider a book described on Semantic Web. It will have links to author, genre, type of book (fiction / non fiction) and some tags. Given the URI of the book, Analogy engine will retrieve the books (URI) which are most similar to it, i.e. those written by same author or having similar keywords. Note that the similarity between two URI is calculated by counting the number of shared links.

In order to perform fast computation of similarity we use a recently invented method called as Analogy Space [1] which can perform Approximate Reasoning and Analogy over semantic networks. Analogy Space uses Singular Value Decomposition over node-feature matrix representation of the semantic network. The node feature matrix is approximated by Sparse Truncated Singular Value Decomposition. Thus Information contained in the node-feature matrix is represented as vectors in low dimensional space. Similarity between two nodes can be found by computing dot product between vectors representing each node. The low dimensional space is called as Analogy Space.[1]

Even though Analogy Space is very useful, it cannot properly approximate a large network. This is due to the fact that semantic networks tend to be modular in nature. The modules have high connectivity inside them. Thus the underlying distribution assumed in Singular value decomposition is not valid while working with matrices representing semantic networks. Using Conceptnet [1] we describe how dividing a semantic network into multiple communities and creating a separate Analogy Space for each module can improve the results.

2. ANALOGY SPACE

2.1 Node Feature matrix representation:



Analogy Space utilizes node-feature matrix representation of the semantic network. The rows of the matrix represent nodes while columns represent features. A feature consists of a combination of a predicate and a node. While creating a feature the directionality of the edge is preserved. Each edge in the network leads to two entries in the matrix. Due to sparse connectivity in semantic networks the matrix is also sparse. The node-feature matrix in normalized & scaled before performing Singular Value Decomposition. [1]

2.2 Approximation using Singular Value Decomposition:

The Node-Feature matrix is approximated using truncated singular value decomposition (SVD). The SVD leads to three matrices. The Matrix U contains rows (vectors) representing nodes in space of the principal components, the matrix S consists of rows representing each feature in space of principal components. The Space of principal components is called as Analogy Space.[1]

$$A_{m \times n} \sim U_{m \times k} \; S_{k \times k} \; V_{k \times n}$$

n = Number of nodes
m = Number of features
k = Number of significant principal components calculated

2.3 Finding analogous nodes using Analogy Space:

In order to find analogous nodes, the row (vector) representing the given node in matrix U is multiplied with matrix U (i.e. dot product with vectors representing all nodes). The result of the multiplication is used as a score to sort and rank the similar nodes.

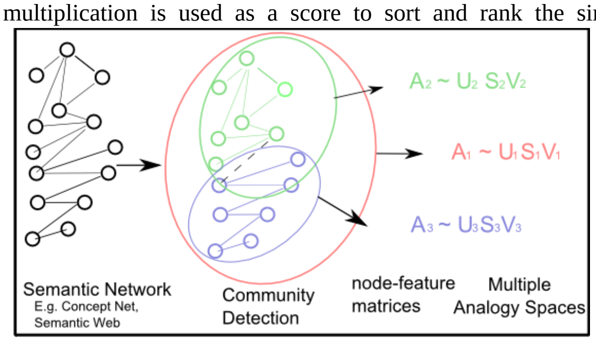


Fig. 2 Multiple Analogy Spaces

3. COMMNITY DETECTION & MULTIPLE ANALOGY SPACES

A number of algorithms have been developed for detecting communities in complex networks. One of the popular algorithms is by Clauset-Newman-Moore (CNM) which performs greedy optimization of a term called as modularity. Modularity measure the quality of the community structure found. [2]

The result of the CNM algorithm is a dendogram of nodes. Any node which has nodes below it is considered as a separate community. Thus a number of communities overlap due to the hierarchical structure produced by the CNM algorithm. [2]

Rather than creating a single node feature matrix for the entire network, nodes appearing in a single community are considered together to create separate node-feature matrix. The community specific node-feature matrix contains rows representing nodes that appear in that community and ALL features (except those which are zero for all nodes in the community). A separate analogy space is created for each node feature matrix by performing SVD.

4. EVALUATION USING CONCEPTNET

We evaluated our procedure on Concept Net, a semantic network describing human common sense knowledge. Conceptnet consists of concepts linked with each other by simple relations such as *part of*, *isa*, *causes* etc. E.g. "Injury *causes* pain". The net work contains around ~14,000 nodes and ~36,000 edges.[3]

Performing Community detection on Conceptnet using CNM algorithm, we found 9 communities each having more than 500 nodes. We calculated 50 significant principal components of all matrices to create multiple Analogy Spaces. [1]

We found that many nodes which were not correctly represented in a single Analogy Space were properly represented in the Analogy Spaces of smaller communities. The overlap of communities leads to different contexts in different communities. In a Space of one community similar nodes for chicken were birds, while in another they were food items.

The Table 1 shows similar concepts for node 'article' in different communities. In the whole network the similar nodes are related to book while in one of the community they are newspaper articles such as Ad and comic. In another community the similar nodes are things such as pamphlet, stylus etc.

Table 1. Qualitative results using multiple Analogy Spaces

Node	Rank	Similar nodes in Analogy Space of		
		Full network	Different communities	
Article	1	Literature	Pamphlet	Ad
	2	Librarian	Mechanica l pencil	Comic
	3	Book	Stylus	Column
	4	Volume	Paper salt shaker	Article
	5	Card catalog	Picture	Opinion

Due to constraint of time, we could not carry out a quantitative user study. However we provide working code for reproducing the results in the companion website.[1] The code can also process networks in RDF format using Rdflib and python.

5 CONCLUSIONS & FUTURE WORK

In this work we have introduced concept of multiple Analogy Spaces to deal with large scale networks. The use of community detection and multiple Analogy Spaces leads to better representation of nodes and uncovers different contexts. The different contexts in different Analogy Spaces are very useful. E.g. while using Analogy Engine with a social network dataset, given a certain person (FOAF profile), Analogy Engine can distinguish between similar persons in his work place and similar persons in his friends circle as both would fall in different communities.

The idea of Analogy Engines for Semantic Web is still in infancy and we hope to refine it by creating an Analogy Engine for Dbpedia using multiple Analogy Spaces. Another area where we are focusing our attention is community detection methods which can detect communities in large Scale networks and algorithms for detection of overlapping communities.

5. ACKNOWLEDGMENTS

We would like to thank members of Common Sense Computing group at MIT Media Lab and Prof. Aaron Clauset for sharing the code and dataset used in this paper.

6. REFERENCES

[1] Speer, R., Havasi, C., and Lieberman, H. 2008. AnalogySpace: Reducing Dimensionality of Common Sense knowledge. AAAI, 2008.

[2] Clauset A., Newman M.E.J. and Moore C., Finding community structure in very large networks. Phys. Rev. E 70, 066111, 2004.

[3] Havasi C, Speer R, and Alonso J. ConceptNet 3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge, RANLP 2007

1 Companion Website: http://www.akshaybhat.com/Aengine/