2012 International Conference on Information and Knowledge Management (ICIKM 2012)
IPCSIT vol.45 (2012) © (2012) IACSIT Press, Singapore

Knowledge Graph Construction for Organizations utilizing Social and Concept Relationships

Nuwan Samarasekera

Global Technology Office, Virtusa (Pvt) Ltd.

Abstract. Mapping knowledge to people in an organization is an important step in enterprise knowledge management. Such mapping aids in identifying resources within an organization to reach out for a given query, identify knowledge gaps and get a better picture of the knowledge spread in the organization allowing for more effective knowledge audits. However, the traditional approach to achieving such is mostly based on asking employees to list down their expertise areas and rate their skill levels for each area in some scale. This approach has many limitations in understanding the complex knowledge dynamics of an organization such as the effect of social connections in knowledge transfer as well as the interrelatedness of concepts. In this paper I explore a mechanism for improving the knowledge mapping in an organization based on a graph representation that utilizes the concept and social relationships.

Keywords: Knowledge Map, Concept Relationships, Term Co-occurrence, Organizational Knowledge Management, Social Relationships

1. Introduction

The traditional approach of asking employees to list down their skills has 2 major drawbacks:

1) Listing down everything a person know is infeasible
2) Typically people list down at a higher granularity, hence fine grained pieces are not mentioned

An effective mapping system should be able to derive an extended knowledge map based on the inputs provided by the users in terms of skill lists. In order to do this, it is essential that the proposed system understands the actual knowledge acquisition and transfer mechanisms.

The traditional system is unable to do the extended inferences about a persons' knowledge because it overlooks 2 main factors:

1) The effect of concept relationships
2) The effect of people relationships

1.1. Effect of Concept Relationships

Concepts are by themselves interrelated. For instance, "Build Automation" is related to the concepts such as "Ant", "Make" etc. More loosely it is related to concepts such as "Automation" and "Software lifecycle". Therefore, for instance, a person who knows about "Build Automation" has a high probability of knowing about "Ant". It is not necessarily the semantic relationship which is of importance in this context (whether there exist an "Is a" or "Part of" relationship between the concepts), but whether the concepts are interrelated in such a way that if a person knows of a concept K1, how likely is he to know of the other concept K2.

There are many researches done in terms of extracting semantic relationships between concepts such as synonym, hyponym, and meronym/holonym based on text analysis [1,2,3] and many published semantic networks of concepts that are widely acclaimed such as the wordNet lexical database. This approach does not suit the problem at hand since being semantically unrelated does not imply that two concepts are not likely to be known together. For instance, consider the keywords “XmlHttpRequest” and “Ajax”. Although the former does not have a clear semantic relationship with the latter, it is obvious that provided that someone knows about “XmlHttpRequest” there is a high probability of him knowing about Ajax. Another difference is that semantic maps do not attempt to distinguish two relationships based on the strength of each and is only interested in the existence or non-existence of a semantic relationship between the two. The weighting is however significant in constructing the knowledge map. For instance, provided that a person knows about “Oracle DB”, although he might know of “PL-SQL” as well as “Hibernate”, the former is a more probable inference than the latter. Therefore, the length of the edge between “Oracle DB” and “PL-SQL” should be lesser than that between “Oracle DB” and “Hibernate”.

1.2. Effect of Social Relationships

The social connections within people in an organization play a tremendous role in the knowledge transfer [4, 5]. For instance, if a person knows a lot of people working in Databases, that person has a better chance of having some insight about databases compared to someone who doesn’t have many contacts in the area. Therefore, in the knowledge map, there should be edges between people who would transfer knowledge to each other. It shall be noted that in those social connections, the edge should represent the probability of knowledge transfer and not just whether 2 people are friends or not. Also noteworthy is that these social connections aren’t necessarily the same as the connections in the organizational hierarchy diagram [6] which leads us to explore different mechanisms than relying on the standard hierarchical links between employees in constructing the knowledge transfer graph.

2. Solution

In order to capture both the effects mentioned above, I construct a graph in which nodes could be either a person or a key-phrase (Concept). The weights for the edges between connected nodes determine the strength of the relationship between the 2 nodes. The meaning of the weights differs based on the type of the nodes of either side, and thus the calculation of the weights. There are 3 types of edges in the graph:

People – People: The weight of the edge from P1 to P2 should indicate the likelihood of P1 knowing a concept K1 provided that P2 knows the concept K1.

Concept- Concept: The weight of the edge from K1 to K2 should indicate the likelihood of a person knowing concept K1 provided that he knows K2

People – Concept: The weight of the edge from person P1 to concept K1 should indicate the likelihood of P1 knowing the concept K1. (This is identical to the skill level measurement provided by users).

The person nodes as well as concept nodes were input to the system by integrating with the FAST Enterprise Search. Further keywords were input by means of analysis of documents within the organization as well as some external documents such as related Wikipedia articles. The documents are first sent through a text extractor in order to remove formatting elements such html tags. Then the extracted text is input to a key word extractor which reports keywords along with the relative strength of each phrase as a key term. I used a keyword extraction implementation based on Matsuo’s work since it does keyword extraction based on a single document while providing comparable results to TF-IDF based mechanisms [7]. This allows for easier scalability since the requirement of scanning the entire corpus is eliminated due its dependency on a single document. Then I filter out low strength keywords, and input the remaining keywords to the graph.

2.1. Calculation of edge weights

2.1.1. Concept – Concept Edges

Although there are general purpose lexical maps such as WordNet [8], it must be noted that such is not possible within the enterprise since those general purpose maps do not cover the organization specific knowledge (projects, tools, processes, internal documentations etc). Also, manual construction of the map is not a feasible option within an organization due to limited resources as well as infeasibility of maintaining such with growing and aging industry specific knowledge. Therefore we need a simple automated mechanism of deriving concept interrelationships.

As mentioned above, there is a link between two concepts in the graph if a person knowing about either concept has a likelihood of knowing the other as well. In order to derive this likelihood, we consider a hypothetical person who acquires knowledge only by reading documents in the organization. The question we have is, provided that a person knows concept K1, what is the likelihood of him knowing concept K2. In the hypothetical experiment, knowing concept K1 corresponds to the person having read documents which have a mention of the concept K1. There are 2 important statistical measurements that we need to consider:

$$TF(K): Term\ Frequency\ of\ K\ in\ corpus$$

$$Cooc(K1, K2): Cooccurrence\ of\ K1\ and\ K2\ in\ corpus$$

Let's define a third function:

$$tf(k, d_i)\ = Term\ Frequency\ of K\ in\ Document\ d_i$$

Then the above statistical measurements can be described as follows:

$$TF(K) = \sum_i tf(K, di)$$

$$Cooc(K1, K2) = \sum_i tf(K2, di)\ \ [\forall i\ tf(K1, di) \neq 0]$$

Consider the hypothetical person in our experiment again. If there is a positive co-occurrence between two words K1, K2, that means a person who has gotten to know K1 through reading those documents have a chance of having read about the word K2. This means higher the co-occurrence, the better is the probability of a person knowing K1 knowing of K2. This makes sense because a document is a coherent piece of knowledge and words that are occurring within the document should be closely related. Therefore we can infer that the strength of arc S(K1,K2) should be positively correlated with Cooc(K1,K2). The usage of Term Co-occurrence in analyzing conceptual relatedness has been established as a successful mechanism by many previous researches as well [9, 10].

Now, let's consider the effect of TF(K2). A higher Term frequency for the word K2 means that the concept is comparatively common in the corpus and is a less specific concept [1, 11, 12]. Therefore, the probability of a person knowing a concept K1 simply because he has read about K2 becomes lesser. For instance, we can't guarantee that a person would know of WCF (Windows Communication Framework) simply because he has read about C#. This is because C# is a too generic a topic and therefore although WCF might have co-occurrences with C#, it is possible that our hypothetical reader got to know of C# through some documents which does not deal with WCF. When the gap between the TF(K1) and Cooc(K1,K2) increases, this probability of a person having read a document related to K1 but haven't met with the concept of K2 increases. Hence, we can make the inference that there should exist a negative correlation between S(K1,K2) and TF(K2).

Therefore we could come up with the following equation to sum up the effects of both statistical measurements in calculating $S(k1, k2)$.

$$S(K1, K2) = Cooc(K1, K2) -\ g(TF(K2))$$

It shall be noted that TF(K) is a faster growing function compared to Cooc(k1,k2). Therefore we need to smoothen the effect of TF(k) by using a function g(x) which needs to be a slower growing function than f(x) = x. Given that the calculation will be done across possibly millions of edges, the function needs to be easily calculable as well. Then we selected the function:

$$g(x) =\ x/\lambda$$

The function satisfies both the requirements of ease of calculation and being a slower growing function. This was compared with other functions such as the square root and the logarithm in terms of accuracy, and our selection proved to be a better choice by the experimentations conducted which is detailed in the Results section.

We have discussed only of the strength of the arc S(k1,k2) so far. The strength thus far discussed is positively correlated with the relatedness. But in the actual graph, the edges should be shorter when they are interrelated and distant when the relationship is weak. i.e.: the edge weight should be smaller as the relationship gets stronger. Therefore we assign

$$W(k1, k2) = \frac{1}{S(k1, k2)}$$

By substituting for the equation derived for S(x) and simplifying, we get:

$$W(k1, k2) = \frac{\lambda}{\lambda * \mathrm{Cooc(k1, k2)} - \mathrm{TF(k2)}}$$

The next question is the selection of value for constant: λ. The important note to be observed is that as TF(k2) reaches λ * Cooc(k1,k2), W(k1,k2) reaches infinity. We use this property in determining the value for λ. We assumed 5% of all the edges derived using the above mechanism of co-occurrence to be noise edges (too distant to be considered interrelated) and experimentally checked which value of λ would result in 5% of the edges reaching infinity. λ =18 provided a good approximation with 5.27% of the edges reaching infinite length.

It shall be noted that as per the above equation, W(K1,K2) ≠ W(K2,K1). This is a required feature which can be understood by revisiting the definition of the weight of the edge. The probability of a person P knowing K1 provided that he knows K2 is not the same as him knowing K2 provided that he knows K1. This can be further clarified with an example. Consider “Build Automation” and “Ant”. While a person who knows of “Ant” must certainly know about “Build Automation” the inverse doesn’t necessarily hold true. i.e.: A person could know of “Build Automation” without knowing about “Ant”. He could have learnt “Build Automation” by using “Make” or “Maven”.

2.1.2. Person – Person edges

The other important edge calculation is that of edges between people. As mentioned in the previous section, the edge weight between person P1, and P2 should correspond to the likelihood of information transferring from P1 to P2. In other words, the edge represents the likelihood of person P2 knowing of a concept K1 provided that person P1 knows of K1. There are 2 statistically important measurements with respect to the calculation of edge strength:

$$fc(P)\colon Friend\ Count\ of\ Person\ P$$

$$mfc(P1, P2)\colon Mutual\ Friend\ Count\ of\ P1\ and\ P2$$

Lets’ consider a hypothetical group of people with each having some friends within the group. Consider the case where a person P1 gets to know of a certain concept K1. He will communicate this information to his friends. The more the number of friends he has, the lesser are the chances him telling it to all of his friends, and thus lesser the probability of a given individual friend of him hearing about it. For instance, if a person had only 2 friends, he would be communicating with those 2 more often thus leading to him transferring his knowledge more completely to the 2. But if he had 100 friends, the chances of him communicating with each and telling the concept becomes lesser. Therefore, it can be inferred that $fc(P1)$ has a negative correlation with S(P1,P2). Similarly, the same can be inferred of $fc(P2)$. This is because, if person P2 has to listen to a lot of friends, then his probability of receiving all information coming from all of his friends becomes lesser.

Now consider the effect of $mfc(P1, P2)$ on the knowledge transfer. A higher $mfc(P1, P2)$ means that there are many friends of P2 who are friends of P1, to whom P1 will be transferring his knowledge. The probability of P2 receiving the communicated knowledge increases when there are more mutual friends, because there are more paths from P1 to P2 for knowledge to transfer. Therefore, we can infer that $mfc(P1, P2)$ has a positive correlation with S(P1,P2). Summing up both the effects, we come up with the following equation for strength of social edges for knowledge transfer:

$$S(P1, P2) = \frac{mfc(P1, P2)}{fc(P1) + fc(P2)}$$

As before, we assign W(P1,P2) to be inverse of S(P1,P2). Thus we get:

$$W(P1, P2) = \frac{fc(P1) + fc(P2)}{mfc(P1, P2)}$$

Unlike the *Concept – Concept* edges, the *Person – Person* edge weights are symmetrical. i.e.:$W(P1, P2) = W(P2, P1)$.

2.1.3. Person – Concept Edges:

Those edges correspond to the level of competence of a person related to the concept. This information is what is input by the users as skill level for each area of expertise. I used the values after normalizing them to fit with the rest of the graph.

2.2. Usage

Once the graph is constructed, we find the best person matches to a given knowledge query by running a shortest path algorithm to find the closest people to the concept in the graph. We used Dijkstra's algorithm since all the edges are positive weighted and used an Adjacency List based graph representation since the graph is sparse with an approximate breadth of 100 (total number of nodes: 10000). We extended the algorithm to handle knowledge queries with multiple concepts included. We first found the closest set of people for each concept. Then we combined the sets and assigned rating for each person based on the following equation:

$$R(P) = \sum_i D(Ki, P) * Avg(TF(K))/TF(Ki)$$

The equation essentially sums up the distances from each word and multiplies with a weight function based on the relative Term Frequency of each word. The objective of this equation is to give priority to people knowing the fine grained topics related over the generic topics. For instance, if a persons' query has "XmlHttpRequest" and "Java", the better fit for the query would be a person who knows of both but in the absence of such, the better would be a person who knows of "XmlHttpRequest" rather than a person who merely knows of "Java".

2.3. Presentation of Graph

The constructed graph was stored in a relational database (MSSQL) and a web service was set up to provide access to the data. The web service would construct an in-memory cache of the graph in order to provide faster access to the graph data. A website was hosted which would enable exploring and searching the graph in a user friendly manner. The website was constructed in html and Javascript, and the graph visualization was done using the d3.js jQuery based library which has good data visualizations as well as high customizability of features. The UI is as below:

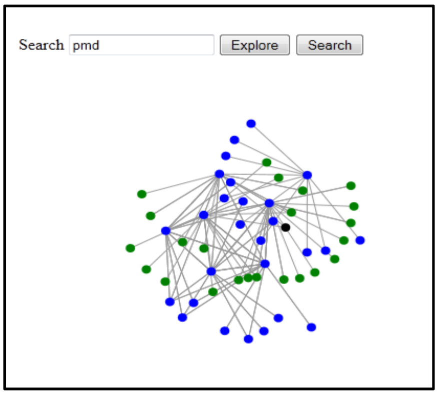

Fig. 1: Visualization of Graph.

Users can enter a keyword in the search box. This is the starting point of the graph exploration. Users are given two options:

1) Explore: In this mode, users can click on a node in the graph, and it would expand the node clicked.
2) Search: In this mode, the graph will continuously expand the closest node found so far and grow until the sufficient numbers of people are reached in the search. Search results are also listed down after the search is completed.

3. Results

Two experiments were conducted to gauge the accuracy of the proposed system. The first was to evaluate the effectiveness of the concept links constructed in the graph. A survey was conducted in which each question consisted of 2 concept links chosen from the database. The participant was asked to choose the concept link which is closer in relationship compared to the other. The questions were formatted in MCQ model to make it easy to answer. A 3rd option was given which could be selected if the participant felt he did not have sufficient knowledge to rate the 2 links. The 2 links were selected randomly in the following manner:

Initially a key phrase is selected from the database at random. Then another phrase out of the closest phrases to the first is selected at random as the second phrase. These 2 words serve as the first concept link. A 3rd phrase is selected out of all the phrases that are connected to the first phrase. This and the first phrase serve as the 2nd concept link. The 2 links are then presented in a random order so that participants would not be able to guess the closer pair by the ordering of the 2 links.

The survey was conducted with the participation of 12 volunteers. The survey gathered 202 expert judgments on the relationship orderings of concept links. Those were compared against the system judgment based on the edge weights assigned. The system showed 75.48% of accuracy overall. The accuracy however was noted to depend based on the length differences of the concept links. The following graph shows the fluctuation of accuracy based on the edge length difference:

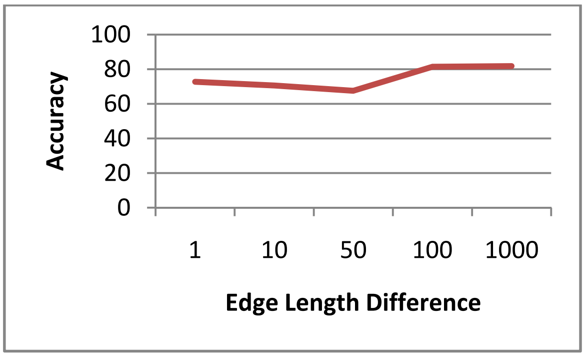

Fig. 2: Accuracy vs. Edge Length Difference.

The system showed higher accuracy when the edge length difference was larger. Afterwards, the strength function S(K1,K2) was altered to compare the performance of different functions in terms of accuracy. The following table demonstrates the accuracy of different strength functions:

Table 1: Comparison of Strength Functions.

Function	Accuracy
$\frac{Cooc(K1,K2)}{TF(K2)}$	53.91%
$Cooc(K1,K2) - \sqrt{TF(K2)}$	73.52%
$Cooc(K1,K2) - log(TF(K2))$	74.01%
$Cooc(K1,K2) - \frac{TF(K2)}{18}$	75.48%
$Cooc(K1,K2) - \frac{TF(K2)}{50}$	73.52%

The second experiment involved in gauging the accuracy of the entire system as a whole. In this experiment we randomly selected 30 word phrases from the database and queried the system for best matching people. We then randomly picked a single person from the top 10 choices listed and asked from the chosen person whether he knows about the particular phrase. Out of the 30 contacted, 22 employees were well aware of the particular area that was questioned. Thus, the system showed 73.33% of overall success rate in finding the experts for a given concept within the organization.

4. Future Work

In the above work I have not distinguished between concept relations and social relations in finding the shortest paths. A possible future direction would be to further explore whether both the relationships equally contribute to the proximity between a person and a concept and if not what differentiation needs to be done. If a differentiation is required this could be embedded in to the edge weight functions as an edge-type based multiplier. We also need to understand how semantic relationships between concepts could be embedded in to the graph. Another possible direction is to figure out how to handle homographs. Such words could result in unrelated concepts becoming more closely connected in the graph via their individual relationships with the homograph.

5. References

[1] M. Sanderson, and B. Croft. Deriving Concept hierarchies from text, in the proceedings of ACM-SIGIR Conference on Research and Development in Information Retrieval, 1999.

[2] M.A, Hearst. *Automated Discovery of WordNet Relations*, in WordNet: an electronic lexical database, Christiane Fellbaum (Ed.), MIT Press, 1998.

[3] G. Grefenstette. *Exploration in Automatic Thesaurus Discovery*, Boston, MA: Kluwer Academic Publisher, 1994.

[4] T. Hansen. *Knowledge Networks: Explaining* Effective *Knowledge Sharing in Multiunit Companies,* in ORGANIZATION SCIENCE Vol 13, NO. 3, May-June, 2002.

[5] G. Szulanski. *The Process of Knowledge Transfer: A Diachronic Analysis of Stickiness. In* Organizational Behavior & Human Decision Processes, Vol. 82, Issue 1, pp9-27, 2000.

[6] R. Cross, A. Parker, and S. Borgatti. *A bird's-eye view: Using social network analysis to improve knowledge creation and sharing*. In Social Network Analysis, published by IBM Institute for Knowledge based organizations, 2002.

[7] Y. Matsuo, and M. Ishizuka. *Keyword Extraction from a Single* Document *using word co-occurrence statistical information,* In the International Journal on Artificial Intelligence Tools

[8] G.A. Miller. *WordNet: A lexical database for english*, in the Communications of the ACM, 38(11): 39-41, 1995.

[9] J.A. Bullinaria, J.P. Levy. *Extracting Semantic representations from word co-occurrence statistics: A computational study*, in Behavior Research Methods.

[10] S. Momtazi, S. Khudanpur, D. Klakow. *A Comparative Study of Word Co-occurrence for Term Clustering in Language Model-based Sentence Retrieval.*

[11] R. Forsyth, R. Rada. *Adding an edge* in Machine Learning: applications in expert systems and information retrieval, Ellis Horwood Ltd: 198-212, 1986.

[12] J.J. Jiang, W. Conrath. *Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy*, In Proceedings of International Conference Research on Computational Linguistics (ROCLING X), 1997.