Cold Start of Enterprise Knowledge Graph Construction

Rong Duan and Kangxing Hu
Huawei Technology, Shenzhen, China

Keywords: Enterprise Knowledge Graphs, Cold Start, Link Prediction.

Abstract: Enterprise Knowledge Graphs(EKG) is a powerful tool for Enterprise Knowledge Management(EKM). Most EKG construction suffers cold start problem. In reality, EKG construction is an interactive process,in which domain experts provide a small seed graph, and data driven methods are applied to expand the graph. This paper proposes a framework to solve EKG cold start problem by integrating graph form expert knowledge with non-graph form corpus. The proposed framework employs expert knowledge to guide unsupervised learning, and crosses check the quality of expert knowledge simultaneously. A coarser cluster level and finer entity level vectorization is proposed to predict the link between graph nodes and cluster words. And also, a combined strategy is adopted to measure the importance of the predicted link, and provide to the expert to evaluate incrementally. The proposed framework solves the "labor intensive" EKG cold start construction problem and utilizes expert knowledge efficiently. Simulation is generated to illustrate the properties of defined measurements, and real-world application is discussed to show the challenges in practices.

1 INTRODUCTION

Basically, there are two types of KG construction techniques: data-driven or manpower driven. Most open-domain KGs are constructed through data-driven method. It either has a large knowledge base at the start point, or has a large corpus that can extract knowledge from. For example, Yago roots from the large knowledge base Wikipedia, WordNet and Geoname(Suchanek et al., 2007). NELL (Carlson et al., 2010) extracts knowledge from hundreds of millions of web pages. For specific domain, the KG construction heavily depends on domain experts. Especially in solving cold start problem. Expert driven method is efficient in a relative narrow area or a field with relative complete ontology. Successful applications such as Amazon product graph(Dong, 2018), medical related knowledge graphs(Ramaswami, 2007)(Ferrucci and Brown, 2011).

Enterprise Knowledge Graph(EKG), as a tool for Enterprise Knowledge Management(EKM), is far more complicated than specific domain applications. The comparison of open domain, specific domain and EKG has discussed in the tutorial(Duan and Xiao, 2019). Usually EKG covers different knowledge domains, and the ontology design is time consuming and labor intensive, which is the bottle neck of EKM. Even though many industries have collected more and more data with digital transformation, the size is still not comparable with open domain.

A big challenge that modern EKM faces is how to utilize the data and extract knowledge from it. Data driven Knowledge Graphs have been receiving more and more attention in enterprise. There are research in studying the EKG framework(Galkin et al., 2017), components and construction steps (Pan et al., 2017), but the cold start problem of knowledge acquisition hasn't been addressed in details yet. This is a problematic issue that holds up the step for enterprise to adopt the knowledge graph approach in EKM.

In practice, EKG starts with a small expert constructed seed graph, and a moderate size corpus can be used to expand the graph. The problem can be decomposed into four subproblems as described following:

1. Validate the correctness of expert designed seed KG.
2. Link the words in cluster with nodes in graph.
3. Integrate expert knowledge with data.
4. Design the predicted link evaluation criteria.

This paper proposes a framework to answer above questions. Our main contributions are three points. First, dispersion and reachability are defined to measure the relation between graph structure and cluster

Duan, R. and Hu, K.
Cold Start of Enterprise Knowledge Graph Construction.
DOI: 10.5220/0010055301530160
In *Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2020) - Volume 3: KMIS*, pages 153-160
ISBN: 978-989-758-474-9

results. This is the first to use graph-form and non-graph-form data to learn from each other, which provides a method to integrate expert's prior knowledge with unsupervised learning. Second, the entities are vectorized by the cluster results, and the link prediction between the graph nodes and words in cluster is decided by the similarities of the coarser cluster level, and the finer entity level. Third, an important-based incremental method is proposed to evaluate the predicted links.

The rest of the paper is organized as follows: Section 2 introduces the related work and problem statement. Section 3 provides definitions and methodology. Section 4 discusses the issues in real-application. Section 5 concludes the paper and gives the future direction.

2 RELATED WORK

Graph-based methods have been used widely in classification and clustering problems. If the data is graph-form, graph properties will apply directly(Rogers and Hahn, 2010)(Shervashidze et al., 2009)(Geng et al., 2012)(Fortunato, 2009). If the data is non-graph-form, a graph will be constructed first, and then apply graph properties(Wang et al., 2005)(Wu et al., 2018)(Zhang and Lee, 2007)(Zhu et al., 2003).

This is different from our problem setting. We have two data sets, one is in graph-form that constructed by experts, and the other is non-graph-form that extracted from corpus.

Another related work is link prediction, especially for cold start link prediction. Traditional link prediction methods base on graph structure or labeled data similarity (Lü and Zhou, 2011)(Martínez et al., 2017)(Taskar et al., 2004)(Menon and Elkan, 2011). Our problem setting is to predict the link between the nodes in graph and words in clusters. The entities are not in the graph yet, the graph structure based link predict doesn't fit. And also, attributes similarity based or labeled data based methods don't fit our short of labels and lack of attributes corpus.

Many steps are involved in constructing KG from scratch. Here we only focus on the stage where experts have manually constructed an initial KG,$G' = \{V', E'\}$, and are willing to expand the G' with the corpus. V' represents node and E' represents edge. V' only has name as its attributes, and E' indicates the existing of relation. The corpus has been segmented to words or phrase(Shang et al., 2017)(Liu et al., 2015), noted as W. The nodes in V', but not in W are noted as V'', The words in W, but not in V' are noted as W''. W'' are candidates of graph nodes. The objective is to expand the original G' with W'' to form a new $G = \{V, E\}$. In general, $|V'| \ll |W|$ for the initial states, where $|V|$ is the number of V. Clustering technology is applied to W to form multiple clusters, represented as C_k, where $k \in [1, K]$, and $|C_k|$ represents the number of words/phrases in the cluster C_k.

3 DEFINITIONS AND METHODOLOGY

3.1 Definitions

Given graph $G = \{V, E\}$, and cluster results C_k, where $k \in [1, \cdots, K]$, dispersion ζ and reachability ρ are defined to measure the relation between graph G and cluster C_k, Cluster-based node vector representation is defined to predict links.

3.1.1 Dispersion

Node dispersion ζ measures the neighbour nodes distribution in clusters. let's denote $\Gamma(V)$ as node V's 1-degree ego-centric network, $|\Gamma(V)|$ represents V's degree. Dispersion of node V is defined as :

$$\zeta_V = \tanh(-\sum_k \frac{p_{\Gamma(V)\in c_k} \log p_{\Gamma(V)\in c_k}}{\log(|\Gamma(V)|+1)}) \tag{1}$$

where $p_{\Gamma(V)\in c_k} = \frac{|\Gamma(V) \bigcap c_k|}{|\Gamma(V)|}$, and $log(|\Gamma(V)|+1)$ is normalization factor to adjust ζ_V 's concentration tendency for lower degree nodes.

ζ_V is tanh transferred normalized entropy. It measures V's neighbors distribution in clusters. The properties of ζ are:

1. $\zeta \in [0, 1]$, due to tanh transformation.
2. $\zeta_V = 0$, when there is only one cluster $K = 1$.
3. $\zeta_V \to 0$, means node V's direct neighbours is relative synchronize with cluster results.
4. ζ_V is monotonic increase function with the number of clusters $\Gamma(V)$ spread into, when cluster method has hierarchical structure.

The properties (1),(2),(3) are straight forward. And the proof of property (4) can be changed to entropy and conditional entropy relation problem, which equivalent to the problem that information gain is nonnegative, which can be proofed easily.

Dispersion ζ measures the synchronousness between graph structure and cluster results from graph point of view. In general, if a node's neighbor is spread in different clusters, its dispersion is high. If the dispersion for most of the nodes in graph is high, it means the cluster result is not in sync with graph structure, the neighbor nodes in graph are spread in different clusters, and the clusters should be combined. The expected dispersion $E(\zeta)$ will decrease as the cluster number decreases until it becomes 0 when every cluster is combined into one cluster.

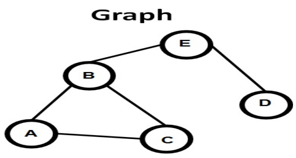


Figure 1: A graph with 5 nodes and 5 edges.

Let's say we have a graph as shown in Figure 1, and cluster method separates the data into 3 clusters: $C_1 = \{A,B,C\}$, $C_2 = \{E\}$, and $C_3 = \{D\}$. For nodes A and C, both of their 1-degree ego-centric network are $\{A,B,C\}$. All of these nodes are in C_1. $\Gamma(A) = \Gamma(C) = \{A,B,C\}$. According to Eq:1, $\zeta_A = \zeta_C = 0$. For node B, its 1-degree ego-centric network is $\Gamma(B) = \{A,B,C,E\}$, 3 of 4 nodes in C_1, 1 of 4 nodes in C_2, and $|\Gamma(B)| = 3$, then $\zeta_B = 0.38$. Same for nodes D and E, their dispersion are $\zeta_D = 0.78$, and $\zeta_E = 0.76$ respectively. Compared with these 5 nodes, A and C's ego-centric nodes are only in one cluster, and their dispersion are 0. B and D's ego-centric nodes are in two clusters, but D only has 1 degree, and B has 3 degree. Adjusted by degree, $\zeta_B < \zeta_D$. Now, let's see the situation when decrease the number of clusters. Combining cluster 2 and 3, the dispersion of all the nodes change to $\zeta_A = \zeta_C = \zeta_D = 0$, $\zeta_B = 0.38$, and $\zeta_E = 0.52$. All the nodes' dispersion become 0, if further combine all clusters into one.

ζ has the tendency to combine clusters. To avoid this situation, cluster reachability ρ_k and expected cluster reachability ρ are introduced.

3.1.2 Reachability

Cluster Reachability. ρ_k is defined by average shortest path of all pairs of nodes in the graph G that overlapped with the cluster k. The nodes that are in G, but not in any cluster are considered when form the shortest path.

$$\rho_k = \frac{\sum(SP(i,j))}{|i,j|}\|_{i,j \in (C_k \bigcap \mathbf{G})} \tag{2}$$

where $|i,j|$ is the number of node pairs. Cluster reachability ρ_k compares the cluster result with graph structure from cluster point of view. It measures the shortest path in graph of each pair of words in cluster k. The smaller the ρ_k, the closer the words are in the graph under this cluster method.

The nodes that in graph, but not in any cluster are considered while calculate reachability, which to avoid disconnected graphs situation.For example, in Figure 1, if node E is not in W, and the cluster results are assumed as $\{\{A,C\},\{B,D\}\}$. The reachability of cluster {A,C} and {B,D} are 1 and 2 respectively,where E is considered in calculating $\rho_{\{B,D\}}$, even though E is not in cluster result. And also, practically, these nodes are the semantic nodes created by domain experts, which help in understanding the semantic meaning of related nodes.

Expected cluster reachability ρ is defined to compare the cluster results with different number of clusters.

Expected Reachability. ρ is the sum of weighted cluster reachability ρ_k for all clusters. The weight is the proportion of number of words in each cluster against number of all words.

$$\rho_C = \sum_k \frac{|C_k|}{|W|}\rho_k \tag{3}$$

The value of ρ represents average steps of separation of the cluster words in graph. The smaller the ρ, the closer the words are in the graph. Referring back to Figure 1, if the clusters are {{A,B,C},{E},{D}}, the cluster reachability are {1,0,0}, and the weights are $\{\frac{3}{5},\frac{1}{5},\frac{1}{5}\}$ respectively. The ρ is 0.6. If the cluster result is {{A,B,C},{E,D}}, $\rho = 1$. If all the nodes are in one cluster, $\rho = 1.7$. The idea situation is that ρ is close to 1, which means the words in each cluster are in one degree ego-centric network in the graph.

Properties of ρ:

1. $\rho \in (0, d(g)]$. where $d(\mathbf{g})$ is the diameter of graph $\mathbf{g}$
2. $\rho = d(\mathbf{g})$,when there is only one cluster K=1.
3. $\rho = 1$, when each cluster is clique, and there is no isolated single nodes.
4. ρ is monotonic decrease with the number of cluster increase, when the cluster method has hierarchical structure.

Proof of property (4): Since the hierarchical clustering result, we can assume the split cluster is s, and $c_s = c'_s + c''_s$. To proof ρ is monotonic decrease with the number of cluster, we only need to proof $\rho_{C_s} \geq \rho_{C_{s'}} + \rho_{C_{s''}}$. The whole proof process is illustrated following.

$$
\begin{gathered}
\rho_{C_s} \geq \rho_{C_{s'}} + \rho_{C_{s''}} \\
\frac{|C_s|}{|W|} \frac{\sum\limits_{|i,j|} SP(i,j)}{|i,j|} \geq \frac{|C_s'|}{|W|} \frac{\sum\limits_{|i',j'|} SP(i',j')}{|i',j'|} + \frac{|C_s''|}{|W|} \frac{\sum\limits_{|i'',j''|} SP(i'',j'')}{|i'',j''|} \\
|C_s| \frac{\sum\limits_{|i,j|} SP(i,j)}{|i,j|} \geq |C_s| \frac{\sum\limits_{|i',j'|} SP(i',j')}{|i',j'|} + |C_s| \frac{\sum\limits_{|i'',j''|} SP(i'',j'')}{|i'',j''|} \\
\frac{\sum\limits_{|i,j|} SP(i,j)}{|i,j|} \geq \frac{\sum\limits_{|i',j'|} SP(i',j')}{|i',j'|} + \frac{\sum\limits_{|i'',j''|} SP(i'',j'')}{|i'',j''|} \\
\frac{\sum\limits_{|i',j'|} SP(i',j')}{|i',j'|} + \frac{\sum\limits_{|i'',j''|} SP(i'',j'')}{|i'',j''|} \leq \frac{\sum\limits_{|i,j|-1} SP(i,j)}{|i,j|-1} - 1 \\
\frac{\sum\limits_{|i,j|} SP(i,j)}{|i,j|} \geq \frac{\sum\limits_{|i,j|-1} SP(i,j)}{|i,j|-1} - 1 \\
\frac{\sum\limits_{|i,j|} SP(i,j)}{|i,j|} \geq \frac{\sum\limits_{|i',j'|} SP(i',j')}{|i',j'|} + \frac{\sum\limits_{|i'',j''|} SP(i'',j'')}{|i'',j''|}
\end{gathered}
$$

As illustrated above, the cluster reachability ρ is equal to 1 when cluster words are direct connect to each other in the graph. ρ has the tendency to separate the data into too many clusters. Neither ρ nor ζ can be used stand alone. For example, in the graph as shown in Figure 2, if the cluster result is {{A,B},{C,F},{E,D}}, the ρ is 1, and the ζ equals to 0.46; If the cluster result is {{A,B,C,F},{E,D}}, the ρ still equals to 1, but the ζ becomes to 0. The reachability of these two results are the same, but the dispersion for the 2nd cluster result is smaller. The second cluster result is better than the first one, since every words in cluster are directly linked to each other in the graph.

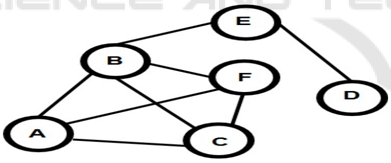
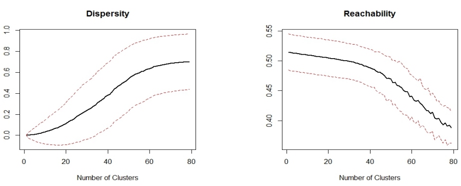


Figure 2: A graph with a 4-nodes clique and 2-nodes clique.

Simulation data is generated to further illustrate the properties of ζ and ρ. A random 1000 nodes erdos-renyi graph is generated with edge probability 0.005. Edge betweeness is adopted as hierarchical clustering method. And the number of cluster is set from 1 to 100. The experiment repeats 50 times. The changes of ζ and ρ according to the number of cluster are shown in the Figure: 3. ζ is monotonic decrease with the number of cluster increase, and ρ is vice-verse.

3.1.3 Entity Vectorization

To predict the link between the nodes in graph with the words in cluster, let's define entity's cluster-based vector representation. Let K represents the number of clusters, and $\overline{\nu}^W$ represents the vector of word W.



(a) (b)

Figure 3: (a) Dispersion monotonic increase with the number of cluster;(b) Reachability monotonic decrease with the number of cluster;Solid black line is mean and dashed red line is variance.

$|\overline{\nu}^W|$ is the length of vector, and $|\overline{\nu}^W| = K$, $\overline{\nu}^W_k$ is the value of kth bit of W.

$$
\overline{\nu}^W_k = \begin{cases} 1, while W \in C_k \\ 0, while W \notin C_k \end{cases}
$$

With this vector representation, the link prediction algorithm is designed as two-steps approach:

1. Cluster-Subgraph Linkage: each cluster C_k will link to a node V_i's ego-centric network $\Gamma(V_i)$, where i = *argmax(JS(*C_k *,* $\Gamma(V_i)$*))*, JS is Jaccord similarity.
2. Cluster Member - Subgraph Node Linkage: each cluster member will link to a graph node where $argmax(\text{cosine}(\overline{\nu}^{C_k^p}, \overline{\nu}^{\Gamma(V_i)_{V_q}}))$. $\overline{\nu}^{C_k^p}$ is the vector representation of entity p in cluster C_k, and $\overline{\nu}^{\Gamma(V_i)_{V_q}}$ represents the vector of nodes V_q that is node V_i's direct neighbour, where V_i is identified in the first step.

3.1.4 Link Validation

Link validation is another challenge problem in KG expansion, especially during the cold start phase.

In EKM system, the initial graph G' is far from complete, and the knowledge consumption base on this graph is poor. The quality validation of the new link is through expert.

To efficiently utilize expert resources under $|V'| \ll |W''|$, this paper proposes a score $\kappa_{W''}$ to measure the importance of each nodes. The score is the average of degree centrality, betweeness centrality and closeness centrality. The nodes with high $\kappa_{W''}$ score are recommended to experts. The validated graph is send back to replace seed graph. The words W'' are added incrementally.

3.2 Methodology

This section describes how to use previous defined measurements to expand *G'* to *G* with *W*. The framework is illustrated in Figure 4. Three key modules are shown in thicker solid rectangle boxes: Cluster Number Determination and Seed Graph Validation, Link Prediction, and Incremental Validation. Other NLP techniques, like Word Segmentation, Phrase Finding, Named Entity Recognition,etc are important, but not in the scope of this paper. We put all these steps in Preprocess box. This framework only considers the stages where the corpus are ready to do cluster, and there exits a small seed graph.

3.2.1 Cluster Number Determination

Cluster Number Determination is the critical module in this framework. Both Link Prediction and Incremental Validation modules base on the result of this one. As discussed in Section 2, we have an initial G' from experts, and a cluster results from clustering method. The largest possible number of cluster is $|N'|$, where $|N'|$ is the number of nodes in initial G'. As illustrated in Section 3, ζ is monotonic decreases, and ρ is monotonic increases. The cross point of ζ and ρ is the best number of cluster. In reality, it is not necessary to start from $|N'|$, just pick the number that doesn't miss the cross point of ζ and ρ. The advantage to use the balance of ζ and ρ to determine the cluster number is that it integrates domain expert's knowledge with data. On one side, these two sources compliment each other with missing information. At the same time, it solves the lack of label, missing attributes and not large enough corpus issues.

3.2.2 Extra Nodes in Graph

There is situation that not all nodes in *G'* are in clusters, which means the entity specified by the expert might not be in corpus. For example, the semantic level entity might not exist in corpus directly. These nodes are represented as V''. To exploit this extra information, V'' needs to be added to clusters. There are two scenarios to add V''.

1. Scenario 1: Add V'' to the cluster that V'' is on the shortest path when calculating ρ_{C_k}.
2. Scenario 2: If V'' hasn't been added in Scenario 1, Add V'' to the clusters that satisfy either of the following two conditions:
   (a) has the most V'''s common neighbours;
   (b) the normalized common neighbor(Jaccard) is larger than a threshold.(We set 50% here).

V'' is critical node for the cluster in the first scenario. For the second scenario, it complements cluster information. The first condition picks the cluster that has the most common neighbor with V'', and the second condition makes sure all those clusters that have a large portion of V'''s neighbors are included. V'' is added to the clusters directly while calculating ρ for scenario 1. V'' is added after the cluster number

3.2.3 Seed Graph Validation

To evaluate the quality of seed graph, the node that is not in any common clusters with its neighbours are selected to re-evaluate by experts. Using Figure 1 as example, if the cluster result is {{A,D}, {B,C}, {B,E}, {D,E}}, $A's$ neighbour $\Gamma(A)$ is B and C. B and C are in clusters $\{B,E\},\{B,C\}$. A is not in any of these two clusters. A's relation with all its neighbours will be re-examined by experts.

$$\forall V' \in G'; if(|V' \bigcap C_{\Gamma(V')}|) = 0, checkV' \Leftrightarrow \Gamma(V')$$

3.2.4 Link Prediction

After determining the number of clusters, the next step is link prediction between graph nodes and cluster members. It is straight forward if there is only one entity from a cluster that exists in the graph. But in most situation, there will be more entities in the graph from the same cluster. Still using Figure 1 as example. If the corpus includes {A,B,C,D,F,G,H,I,K}, and are grouped into four clusters: C_1 :{A,F,G}, C_2: {A,B,K,H}, C_3: {C,D,H,I} and C_4: {B,F}. The relation between cluster member and graph can be summarized into 4 scenarios:

- **Scenario 1:** Only one cluster member exists in the graph, and the other non-graph members only associate with that specific cluster
- **Scenario 2:** Only one cluster member exists in the graph, and the other non-graph members might exist in different clusters
- **Scenario 3:** There are multiple cluster members exist in the graph, and the other non-graph members only associate with that specific cluster
- **Scenario 4:** There are multiple cluster members exist in the graph, and the other non-graph members might exist in different clusters

In the above example, G belongs to scenario 1, F belongs to scenario 2, K and I belong to scenario 3 and H belongs to scenario 4.

To solve the link prediction problem to cover these 4 scenarios, we follow the method in Section 3.1.3, which finds the coarser cluster level linkage first, and finer individual entity level linkage second. The result for the cluster level linkage is shown in Table:1

Node E is added to the cluster $\{C,D,H,I\}$ first according to Section 3.2.2, where E is on the shortest path between C and D. Table: 1 provides the

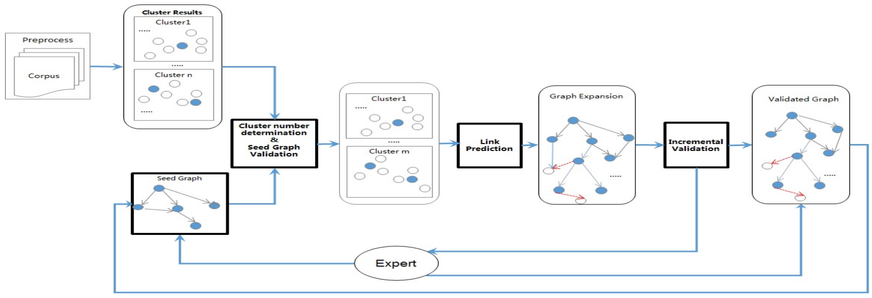


Figure 4: Framework components: rectangle boxes are modules, and round boxes are results. Solid dots represent entities from expert, and empty dots represent entities from Corpus. Thicker solid rectangle boxes are the key components. After important predicted links are validated, the validated graph is send back to replace Seed Graph, and incrementally expand the graph.

Table 1: Link Prediction: Cluster Alignment.

Cluster Members	New Clusters	ego-centric Networks	$JS(G',C_k)$
AFG	AFG	**A**BC	1/5
ABKH	ABKH	**A**BC	2/5
ABKH	ABKH	A**B**CE	1/3
CDHI	CDHI**E**	AB**C**	1/7
CDHI	CDHI**E**	**D**E	2/5
CDHI	CDHI**E**	B**E**D	1/3
BF	BF	A**B**CE	1/5

detail calculation. Both cluster $\{A,F,G\}$ and cluster $\{A,B,K,H\}$ match with A 's ego-centric network $\{A,B,C\}$, cluster $\{C,D,H,I\}$ matches with E's ego-centric network $\{B,D,E\}$, and cluster $\{B,F\}$ matches with D's ego-centric network $\{D,E\}$. The sub-graph and cluster matching results are shown in Figure: 5. The solid nodes are the nodes in graph, and dashed nodes are the candidate nodes that to be added to the graph. The color indicates the coarser level cluster alignment.

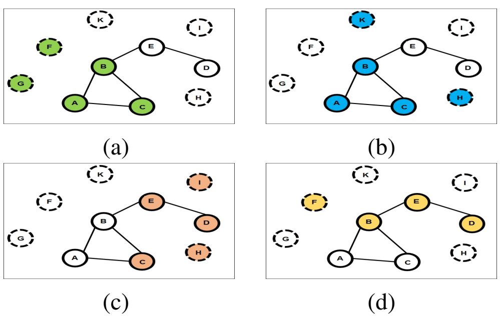


Figure 5: Solid circles are notes in graph, and dotted circles are candidate entities. The same color represents the same alignment. (a) Cluster $\{AFG\}$ aligns with A's ego network $\{ABC\}$; (b) Cluster $\{ABKH\}$ align with A's ego network ABC; (c) Cluster $\{CDHI\}$ align with E's ego network $\{BDE\}$;(d) Cluster $\{BF\}$ align with D's ego network $\{DE\}$.

Table 2: Link Prediction: Entity Linkage.

	A 1100	B 0101	C 0010	D 0010	E 0001
H-0110	0.5	0.5	0.71	0.71	0.71
K-0100	0.71	0.71	0		
I-0010		0		1	1
F-1001	0.5	0.5	0		0
G-1000	0.71	0	0		

After the cluster alignment, the second step of link prediction is to link the individual entities. Since H is in both clusters $\{A,B,K,H\}$ and $\{C,D,H,I,E\}$, and these two clusters link to B's and E's ego-centric network, $\{A,B,C,E\}$ and $\{B,E,D\}$ respectively. H needs to compare the similarity with all the existing nodes A,B,C,D, and E. K only exists in cluster $\{A,B,K,H\}$, and $\{A,B,K,H\}$ has larger Jaccord Index with A's ego-centric network $\{A,B,C\}$, comparing with B's ego-centric network $\{A,B,C,E\}$. K only needs to compute the cosine similarity with nodes A,B,C. Follow the same logic, I needs to compare with B,E and D, F needs to compare with A,B,C and E. The computed cosine similarities are shown in Table 2, and the graph after the link prediction is shown in Figure:6

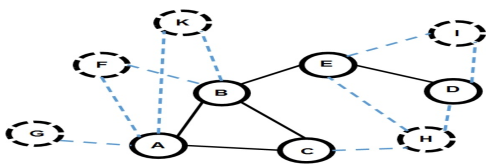


Figure 6: Dashed line are predicted link.

3.2.5 Incremental Validation

After adding W'' to G', the expert evaluation is needed to confirm the expansion. When $|V'| \ll |W''|$, the criteria to recommend W'' to be evaluated is important. We adopt the incremental method to recommend to

Table 3: Procedure to extend G' with W''.

Procedure of expand G' with W
Initial:
1: Input expert designed graph $G' = \{V', E'\}$
2: Initial number of cluster $K = \lvert V' \rvert$,
Find the best cluster number K:
3: While $K >= 2$
4: Cluster W into C_k, k $\in$ [1,K] with hierarchical cluster method
5: for each node $V' \in G'$
6: calculate dispersion $\zeta_{V'}$
7: for each topic C_k, $k \in [1, K]$
8: calculate reachability ρ_{C_k}
9: if V'' is on the shortest path while calculating ρ_{C_k}
10: add V'' to C_k
11: if $\mu(\zeta_{V'}) <= 0.2$, and $\rho \approx 1$, exit;
12: else K = K - skip, go back to Step 3
13: The best K is the cross point of $\zeta_{V'}$ and ρ_{C_k}
14: if V'' is not in C
15: addV(V'', C)
Seed Graph Validation G':
16: for each $V'_i \in V'$
17: if $\lvert V'_i \bigcap C_{\Gamma(V'_i)} \rvert = 0$
18: output V'_i and $\Gamma(V'_i)$ for re-evaluation
Expand G' with W''
19: vectorlize V' and W base on cluster result from Step 16
20: for each $V'_i \in V'$
21: calculate $J_{V'_{ik}}$ between $\Gamma(V'_i)$ and C_k
22: find $J_{V'_{ik}} > 0.5$ or $argmax_{k \in [1,K]}(J_{ik})$
23: for each k in Step 19
24: calculate $cosine \overline{V}_{V'_i}, \overline{V}_{W_j}$, where $V'_i \in \Gamma(V'), W_j \in C_k$
25: find $argmax_{cosine \overline{V}_{V'_i}, \overline{V}_{W_j}}$, add link between W_j and V'_i
Evaluate W''
26: for each W'' added to graph
27: if form a cycle for relation forbidden cycle, output W''
28: else calculate $\kappa_{W''}$
29: Sort $\kappa_{W''}$ from high to low for validating.
30: form new graph G after validation and send back to Step 1.
function addV(V'',C):Add V'' to Clusters:
1: for each $V''_i \in V''$
2: calculate $J_{V''_{ik}}$ between $\Gamma(V''_i)$ and C_k
3: find $J_{V''_{ik}} > 0.5$ or $argmax_{k \in [1,K]}(J_{ik})$
4: Add V''_i to C_k

experts. First the degree centrality,closeness centrality and betweeness centrality are combined to rank the candidate W''. Recommend W'' to experts that have its combined score larger than a threshold. After expert validation, the new graph is send back to replace the seed graph. The whole process repeats till all W'' have been added to G or all W'' 's combined score is smaller than threshold.

The summarized procedure is illustrated in Table 3.

4 EXPERIMENTS

The proposed framework has been tested on an Enterprise Finance Knowledge Management System. The corpus are collected from official documents and internal community discussion blogs. Rules, regulations, policies, and manually collected Q&A regarding accounting, tax, capital, etc are covered. The objective is to construct multiple intelligent bots that can answer questions or serve as searching engine. The detail is removed to reserve the company's property. The procedure will be discussed through the graph properties without loss generality. The size of corpus is few Gigabytes, and the initial expert designed seed graph has few hundred nodes, and most in hierarchical relation. The density of the seed graph is 0.031. We can use HAC to get the monotonic increase and decrease ζ and ρ here, but the character similarity based clustering method lacks of latent topic, which doesn't fit well for our situation. We adopt LDA model in practice. Even though LDA is not strict hierarchical clustering method, the cluster result will form a hierarchical-like result if the corpus has good hierarchical structure. Even though ζ and ρ will not have increase or decrease function monotonically, but the increase and decrease trend is good enough. We set the maximum topic number as the half of the seed graph nodes, and gradually reduce the topic number till to one. Dispersion ζ decrease slowly with the number of topics reduce till a change point. After the change point, ζ drops quickly, which means graph structures are more concentrate in clusters. Reachability ρ has the trend to increase with the number of topics decrease, but it is not monotonic. There are situations that $\rho_k \leq \rho_{k+1}$. Globally, ρ increases with the number of topics decrease. In practice, we use the trend of ρ instead of the actual line to find the cross point with ζ. The changes of dispersion and reachability with the number of cluster for the first round is shown in Figure: 7.

After validation, the final graph has around few thousand entities, with density as 0.23. After the cold start, the base graph serves different bots, and expanded through the usage. The whole EKM system has been running for one year, and have accumulated close to million entities.

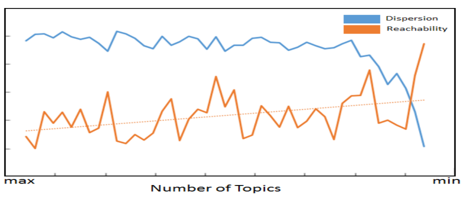


Figure 7: Dispersion and Reachability for real application.

There are some issues we would like to address here in practice. First, In enterprise, the corpus have low frequency and high precision, which is a challenge for frequency based data-driven method; Second, the construction of seed graph can adopt the Enterprise Architecture, which provides a comprehensive view of business; and the last, expert knowledge needs crowd sourcing too, especially when the knowledge are cross different domains. And the validation of expert driven crowd sourcing is another challenge.

5 CONCLUSION

This paper proposes a framework to solve enterprise knowledge graph cold start problem. The proposed framework takes the advantage of graph-formed expert knowledge, and use it to guide the clustering method, which integrates expert knowledge with data driven clustering.

Coarser cluster level linkage, and finer entity level linkage are adopted base on entity vectorization. Incremental validation is used to gradually add the words in graph. The framework is validated through an enterprise finance knowledge graph to support reasonable knowledge consumption. Search engines and Q&A apps will be constructed base on this graph, and the current available data-driven techniques will be used to expand and fine tune the graph.

REFERENCES

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E. R. H., and Mitchell1, T. M. (2010). Toward an architecture for never-ending language learning.

Dong, X. L. (2018). Challenges and innovations in building a product knowledge graph. In *24th ACM SIGKDD International Conference on Knowledge Discovery &Data Mining*, pages 2869–2869.

Duan, R. and Xiao, Y. (2019). Enterprise knowledge graph from specific business task to enterprise knowledge management. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM ’19, page 2965–2966.

Ferrucci, D. and Brown, E. (2011). Adaptwatson: A methodology for developing and adapting watson technology. *IBM, Armonk, NY, IBM Res. Rep., RC25244*.

Fortunato, S. (2009). Community detection in graphs. *Physics Reports*, 486(3):75–174.

Galkin, M., Auer, S., Vidal, M.-E., and Scerri, S. (2017). Enterprise knowledge graphs: A semantic approach for knowledge management in the next generation of enterprise information systems. In *International Conference on Enterprise Information Systems (ICEIS)*, volume 2, pages 88–98, Porto, Portugal. SciTePress.

Geng, L., Semerci, M., Yener, B., and Zaki, M. J. (2012). Effective graph classification based on topological and label attributes. *Statistical Analysis & Data Mining the Asa Data Science Journal*, 5(4):265–283.

Liu, J., Shang, J., Wang, C., Ren, X., and Han, J. (2015). Mining quality phrases from massive text corpora. In *2015 ACM SIGMOD International Conference on Management of Data*, pages 1729–1744. ACM.

Lü, L. and Zhou, T. (2011). Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170.

Martínez, V., Berzal, F., and Cubero, J.-C. (2017). A survey of link prediction in complex networks. *ACM Computing Surveys (CSUR)*, 49(4):69.

Menon, A. K. and Elkan, C. (2011). Link prediction via matrix factorization. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 437–452. Springer.

Pan, J. Z., Vetere, G., Gomez-Perez, J. M., and Wu, H. (2017). *Exploiting Linked Data and Knowledge Graphs in Large Organisations*. Springer Publishing Company, Incorporated, 1st edition.

Ramaswami, P. (2007). A remedy for your health-related questions: health info in the knowledge graph.

Rogers, D. and Hahn, M. (2010). Extended-connectivity fingerprints. *Journal of Chemical Information & Modeling*, 50(5):742–754.

Shang, J., Liu, J., Meng, J., Xiang, R., Voss, C. R., and Han, J. (2017). Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge & Data Engineering*, PP(99):1–1.

Shervashidze, N., Vishwanathan, S. V. N., Petri, T., Mehlhorn, K., and Borgwardt, K. M. (2009). Efficient graphlet kernels for large graph comparison. In *12th International Conference on Artificial Intelligence and Statistics*, pages 488–495.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). Yago: a core of semantic knowledge. pages 697–706.

Taskar, B., Wong, M.-F., Abbeel, P., and Koller, D. (2004). Link prediction in relational data. In *Advances in neural information processing systems*, pages 659–666.

Wang, J., Zhang, Z., and Zha, H. (2005). Adaptive manifold learning. In *Advances in neural information processing systems*, pages 1473–1480.

Wu, X., Zhao, L., and Akoglu, L. (2018). A quest for structure: Jointly learning the graph structure and semi-supervised classification. In *27th ACM International Conference on Information and Knowledge Management*, CIKM ’18, pages 87–96. ACM.

Zhang, X. and Lee, W. S. (2007). Hyperparameter learning for graph based semi-supervised learning algorithms. In *Advances in neural information processing systems*, pages 1585–1592.

Zhu, X., Ghahramani, Z., and Lafferty, J. D. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In *20th International conference on Machine learning (ICML-03)*, pages 912–919.