Implicit Relation Inferring Using Knowledge Graph

A Probabilistic Approach

ABSTRACT

In this paper, we propose an approach for detectiing the implicit relations between 2 entities.

1. INTRODUCTION

2. RELATED WORKS

3. FRAMEWORK

First,we do conceptualization
Next,Judge whether the 2 entities are conceptually same
Then, there are 2 cases of the CanBeExplained function:

- Explain 2 conceptually similar entity

Table 1: conceptually similar entity

entity	concept
Steve jobs	Person
Bill Gates	Person

- Explain 2 conceptually different entity

Table 2: Add caption

entity	concept
Mona Lisa	Painting
Renaissance	Period

Note that the concept here are not unique.

Last, We rank all the explanations in each step.

4. PROBABILITY RECALCULATION AFTER CONCEPTUALIZATION

Given an entity e, from Probase, we can acquire its concepts' set C and for each $c_i \in C$, the frequency $n(c_i, e)$ can be accordingly derived, which means how many times the e isA c_i pattern can be observed from the original corpus.

However, the concepts here has various forms as illustrated in Example 1. For our task, we only need relatively general concepts. The number of entities can be very large, but the number of top concepts and the relationship between them are limited, literally, we can find all the possible relationship between concepts instead of store all the long-tailed entities and their relations, which indicates the rationality of doing conceptualization.

EXAMPLE 1 (VARIOUS FORMS OF CONCEPTS). *Take the entity* `Mona Lisa` *as example, its concepts includes* `painting, famous painting, world's most famous painting`*, with corresponding frequency* `33,8,1`

4.1 Problem Definition

Given *Probase*, for each entity e, we can derive $P(c_i|e)$, where $c_i \in C_{probase}$,

$$P(c_i|e) = \frac{n(c_i, e)}{n(e)}$$

We divide the $C_{Probase}$ into 2 parts, C_{simple} and C_{long}, where C_{simple} only contains one word and C_{long} are the rest. C_{simple} are generated by the head modifier detection. The problem here is to recalculate the probability $P(\gamma_i|e)$ where $\gamma_i \in C_{simple}$, literally, we should contribute all the counts of C_{long} to C_{simple}.

4.2 Problem Solution

After head modifier detection, we have a set of $\gamma_i \in C_{simple}$, among all the $c_{long_j} \in C_{long}$, there are 2 cases in the probase determined by whether the c_{long_j} has an isA edge towards γ_i or not. The intuition of doing so is illustrated in the example 2:

EXAMPLE 2 (CONTRIBUTING LONG CONCEPTS). *Assume that* `Mona Lisa is a painting` *and* `Mona Lisa is a famous painting` *are observed respectively* `33 times and 8 times` *from different documents, we will get the knowledge that* `Mona Lisa is a painting` *occurs* `41 times` *instead of* `33 times`*. There are less chance of occurring* `Famous painting is a painting` *so that there won't be necessarily an isA edge from* `famous painting` *to* `painting`.

Therefore, to calculate $P(\gamma_i|e)$, there are three cases:

1. e isA γ_i. The entity has has an isA edge towards one or more simple concept, which gives the original $P_{original}(\gamma_i|e)$

2. e isA c_{long} , c_{long} isA γ_i, In this case, we need to calculate the following equation

$$P(\gamma_i|e) = \sum_{c^*_{long} \in C_{long}} P(\gamma_i|c^*_{long}, e) \times P(c^*_{long}|e)$$

   , where $P(c^*_{long}|e)$ can be obtained from $Probase$ and

$$P(\gamma_i|c_{long}, e) = \frac{n(\gamma, c_{long}, e)}{n(\gamma_i, e)} \quad (1)$$

   We assume that the occurrence of e does not affect $P(\gamma_i|c_{long})$ equivalently speaking, $P(\gamma_i|c_{long})$ is independent from e, thus Eq. 1 can be simplified

$$P(\gamma_i|c_{long}, e) = P_{probase}(\gamma_i|c_{long}) = \frac{n(\gamma, c_{long})}{n(\gamma_i)}$$

   which can be obtained from $Probase$.

3. e isA c_{long} , c_{long} has no edge towards γ_i. The edge here refers to the isA relationship in $Probase$. Example 2 pointed out that there won't be necessarily an isA edge from `famous painting`(c_{long}) to `painting`(γ_i), however c_{long} obviously belongs to γ_i. In this case, since it's detected by the head modifier method, we assume

$$P_{head}(\gamma_i|c_{long_j}) = 1$$

   Another reason why we do head modifier detection here is that even if the long concept c_{long} has an isA edge towards a certain concept γ_i', it still sometimes not include the head concept of the long concept which is very plausible. The tradeoff of the 2 method is described in Example 3

Notice that the boundary between case 2 and case 3 are not strict, there are such edges that have low observation in Example 3. So that if we consider them as a whole, we can derive:

$$P(\gamma_i|c_{long}) = \lambda P_{head}(\gamma_i|c_{long}) + (1-\lambda) P_{probase}(\gamma_i|c_{long}) \quad (2)$$

where λ is a parameter principle: related to plausibility, number of occurrence, varies for different c_{long} should it be derived from learning ? since we assume $P_{head}(\gamma_i|c_{long})$ to be 1, Eq. 2 is simplified to:

$$P(\gamma_i|c_{long}) = \lambda + (1-\lambda) P_{probase}(\gamma_i|c_{long})$$

EXAMPLE 3 (HEAD CONCEPTS VS ORIGINAL CONCEPTS). *Again take* `famous painting` *as example, whose concepts* `image, treasure` *are reasonable but implausible, since their occurrence are twice and once respectively. However, the most plausible concept* `painting` *is not among the concepts. On the other hand, there exists several concepts that have also reasonable. For example* `topaz`*(a kind of yellow gemstone) has the concept* `precious stones`*, and* `precious stones` *has an edge towards* `material` *which is reasonable.*

Finally $P(\gamma_i|e)$ is calculated using the following equation:

$$P(\gamma_i|e) = P_{original}(\gamma_i|e) + \sum_{c^*_{long} \in C_{long}} [\lambda^*_i + (1-\lambda^*_i) P(\gamma_i|c^*_{long})] \times P(c^*_{long}|e) \quad (3)$$

The process of calculation is illustrated in the example 4

EXAMPLE 4 (CALCULATING $P(\gamma_i|e)$). *As illustrated in Fig. 4.2, the process of calculating the typicality a concept is as follows, where* `painting` *is* γ_i *and* `Mona Lisa` *is* e $P($`painting`$|$`MonaLisa`$)$ *consists of 2 parts, the direct edge* $P_{original}(\gamma_i|e) = 0.23$*, and the second part*

$$\sum_{c^*_{long} \in C_{long}} [\lambda^*_i + (\alpha^*_i) P(\gamma_i|c^*_{long})] \times P(c^*_{long}|e)$$

$(\alpha^*_i + \gamma^*_i = 1)$ *Thus we get*

$$P = 0.007 \times \lambda_{i2} + 0.05 \times \lambda_{i1} + 0.04 \times (\lambda_{i3} + 0.65\alpha_{i3})$$

For `piece`*, it is the similar process. The relation here is only part of the whole graph.*

We consider only 2 layers of isA relationship for 2 reasons. The first one is that more layers will lead to noisy concepts such as `issue, factor, element`, which are concepts for almost eveything, Secondly, discussing the transitive relation between concepts is beyond the scope of this paper.

5. FIND ALIAS FOR ATTRIBUTES

For a pair `(Sherlock holmes, United Kindom)`, `country` is a merely-ok attribute, on the contrary, `residence, deathPlace` are better since they are more specific and more seemingly plausible to be an attribute.We argue that for each pair of entity, there is a selectional preference for attribute.

5.1 Problem Definition

Given a set of concept pairs (γ_1,γ_2), where $\gamma_1 \in C_1$ and $\gamma_2 \in C_2$, we want to find a set of attributes A, where for each $a \in A$: we can form a (γ_1,a,γ_2) pair which best describe the relationship between γ_1 and γ_2.

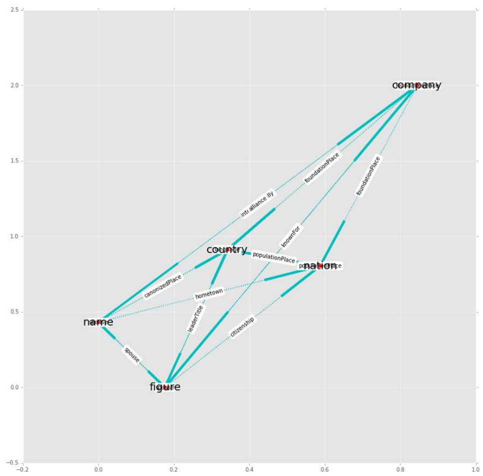

Figure 2: Subgraph of Entity Attribute Graph

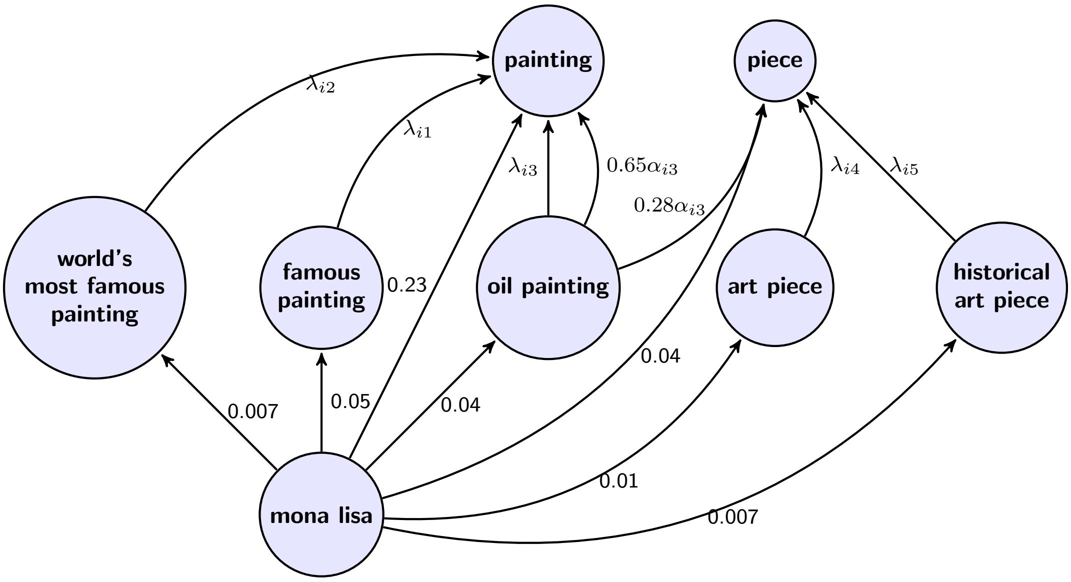

Figure 1: calculating $P(\gamma_i|\texttt{MonaLisa})$

5.2 Problem Solution

5.2.1 *Entity Attribute Graph Construction*

For any $(entity_1, attribute, entity_2)$ tuple, later denoted as (e_1, a, e_2) , where e_1 and e_2 are also referred to as `domain` and `range` of the attribute. We can conceptualize e_1 and e_2 using the method in section 4, and get a set of concept C_1, C_2, accompanied with a set of probabilities $P(\gamma_1|e_{1i})$, $P(\gamma_2|e_2)$,where $\gamma_1 \in C_1, \gamma_2 \in C_2$.

Thus for any attribute a, given a pair of entity (e_{1i}, e_{2j}), we can define: should i use joint ratio here?

$$P_{(e_{1i},e_{2j})}((\gamma_1,\gamma_2)|a) = P_{before}(\gamma_1|a) \times P_{after}(\gamma_2|a) = P(\gamma_1|e_{1i})P(e_{1i}|a) \times P(\gamma_2|e_{2j})P(e_{2j}|a) \tag{4}$$

where we use $P_{(e_{1i},e_{2j})}((\gamma_1,\gamma_2)|a)$ to denote observing a single pair (e_{1i}, e_{2j}), how likely is a combination of (γ_1, a, γ_2) to occur.

Consequently,

$$P((\gamma_1,\gamma_2)|a) = \sum_{e_{1i}\in E_1, e_{2j}\in E_2} P_{(e_{1i},e_{2j})}((\gamma_1,\gamma_2)|a) \tag{5}$$

where E_1, E_2 denoting the whole set of domain entity and range entity,The $P(e_{1i}|a)$ and $P(e_{2j}|a)$ here has only 2 values 1 and 0, depending on whether e_1 occurs before a or e_2 occurs after a. Apparently, only (e_{1i}, a, e_{2j}) occurs will give the equation a non-zero value, therefore, Eq. 5 is finally equal to Eq. 6.

$$P((\gamma_1,\gamma_2)|a) = \sum_{(e_{1i},a,e_{2j})\in KB} P_{(e_{1i},e_{2j})}((\gamma_1,\gamma_2)|a) = \sum_{(e_{1i},a,e_{2j})\in KB} P(\gamma_1|e_{1i}) \times P(\gamma_2|e_{2j}) \tag{6}$$

The process of calculating is demonstrated in Example. 5

Example 5 (Calculating $P((\gamma_1,\gamma_2)|a)$). *As illustrated in Fig. 3, the process of calculating $P((\gamma_1,\gamma_2)|a)$ is as follows*

insert a graph

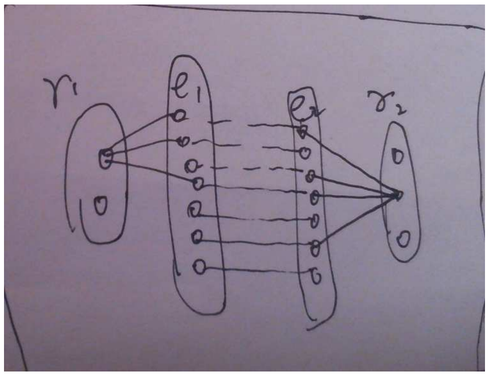

Figure 3: Calculating $P((\gamma_1,\gamma_2)|a$ for

To Construct the Entity Attribute Graph, we calculate $P((\gamma_1,\gamma_2)|a)$ for each attribute.

Note that we only consider the attributes whose range is an entity, and ignore those numerical values or date-and-time values such as $(MonaLisa, Year, 1503)$.

For each (γ_1, a, γ_2) tuple, we can calculate $P((\gamma_1,\gamma_2)|a)$ for each

5.3 Find the best alias

We then Use an $\arg\max$ model to solve the problem.

Given (e_1, e_2), our goal is to find the best attribute for it. We denote it as:

Normalize and get topK, K trough case study is around 5, so we here set K=10

Add edgeïijŽC1->C2 with label: attr, score to the alias graph

6. EXPERIMENT

6.1 Head Concept Vs Original Concept

6.2 Find alias

6.2.1 compare

Compare $P((\gamma_{1i}, \gamma_{2i}|a))P(\gamma_{1i}|a) \times P(\gamma_{2i}|a)$

6.2.2 Sense Disambiguation

We can solve the problem of sense disambiguation problem well by applying this method since there are many entities belongs to the same concept and we only consider topK (γ_1, γ_2) pairs that has high typicality $P((\gamma_1, \gamma_2)|a)$, so that the weird (γ_1, γ_2) patterns as manifest in Example. 6 can be easily filtered.

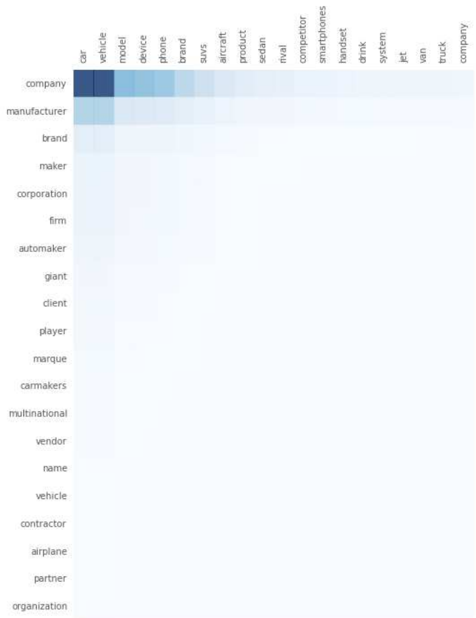

Figure 4: γ_1, γ_2) plot for attribute `Manufacturer`

Example 6 (Sense Disambiguation). *Consider the following $(e1, a, e2)$ tuple* `(iphone, manufacturer, apple)`*. Suppose it is our query, where* `apple`*'s sense can either be a kind of* `fruit` *or a* `company`*. Fig. 4 is a heatmap for all the concepts pairs (γ_1, γ_2) of attributes* `manufacturer`*. The horizontal axis represents the e_1 and the vertical axis stands for e_2. The darker the blue is, the higher typicality it will be. In Fig. 4, We can observe that the top concepts of e_2 in the heatmap are* `company, manufacturer,...` *and top 10 pairs also does not include* `fruit`*. The intuition for this is that there exists thousands of $(e1, a, e2)$ tuple such as* `(BMW_Z4,manufacturer,BMW),(PlayStation_4,manufacturer,Sony)` *other than* `(iphone, manufacturer, apple)` *tuple, which results in a reasonable distribution.*

6.3 Selectional Preference

6.4 Evaluation

7. CONCLUSION