Entity Resolution using Co-occurrence Graph and Continuous Learning

Anoop Kumar Pandey
International Institute of Information Technology
Bangalore
26/C Electronics City
Bangalore, India - 560100
anoopmis@gmail.com

Srinath Srinivasa
International Institute of Information Technology
Bangalore
26/C Electronics City
Bangalore, India - 560100
sri@iiitb.ac.in

ABSTRACT

In a community setting, *Utilitarian Knowledge* or "Knowledge that works" are routinely diffused through social media interactions. The aggregation of this knowledge is a divergent process, where common knowledge gets segregated into several local worlds of utilitarian knowledge. If the community as a whole is coherent, these different worlds end up denoting different aspects of the community's dynamics. To capture and represent this knowledge, several data models have been proposed. One of the model organizes concepts (atomic or simplicter elements) in a hierarchy namely concept hierarchy ("is-a") in which concepts are added manually at the most appropriate level inside the hierarchy. To minimize manual intervention in entity resolution, this paper proposes entity resolution based on co-occurrence graph and continuous learning, thereby eliminating the bottleneck of manual concept entry. While traditional Supervised Learning methods require sufficient training data before hand which is not available in a community setting at start, Continuous Learning method could be useful which can acquire new behaviours and can evolve as the community data evolves.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Web Science**]: Metrics—*performance measures*

General Terms

Algorithms, Performance

Keywords

Knowledge Base, Co-occurrence Graph, Utilitarian Knowledge, Continuous Learning

IWAIT '15, Oct. 8 – 10, 2015, Aizu-Wakamatsu, Japan.

1. INTRODUCTION

In on-line communities, several netizens interact and exchange or share large amount of knowledge among themselves. The knowledge, they share, comes in two flavours: Encyclopaedic or Informational Knowledge & Utilitarian Knowledge that can be put to use. For maintaining Encyclopaedic knowledge many readymade ontologies like DBpedia are available, however there is no such knowledge base to capture Utilitarian Knowledge. To capture and represent the same, a data model called Many Worlds on a Frame (MWF)[4] was proposed. The data model contains several concepts organized in hierarchies. Concept in general, could refer to all the terminologies and vocabulary of a particular domain which is used to describe it. The definition of concepts and relationship between the concepts is typically captured using a simple structure called concept tree that captures two kinds of relationships: 'is-a' and 'is-in'. For instance, the concepts 'Java' **is-a** 'Programming Language' depicts 'is-a' relationship while "Bangalore" **is-in** "Karnataka" depicts containment. These hierarchies contain many concepts beyond the encyclopaedic concepts which are relevant to that domain. Therefore in such kind of knowledge base, concepts need to be added manually and hence poses a bottleneck in evolution of the knowledge base. A need for an entity resolution system, that could, given a concept, label it with proper 'is-a' parent thereby placing it appropriately in the concept hierarchy, could help the knowledge base to grow with minimal manual intervention. While conventional Supervised Learning methods require sufficient training data before hand which is not available in a community setting at start, Continuous Learning method could be useful which can acquire new behaviours and can evolve as the community data evolves.

2. APPROACH

We formulate the problem of finding 'is-a' parent in community knowledge base as a class labelling task. Considering co-occurrences to be the only observed facts in the community posts, we wish to generate a signature for each class(is-a label) using co-occurrence graph. While the co-occurrences between terms in a document represent the associations between terms, signature for a class 'C' represents a vector with co-occurring classes as dimensions and their average co-occurring frequency with 'C' as the value along that dimension . The knowledge base, would then predict 'is-a' parent for an unknown concept by comparing its co-occurrence neighbours with the class signatures using cosine similar-

ity and shall recommend the label having highest similarity. The signatures will be improvised upon user feedback and as when a new instance of that class is added to knowledge base.

2.1 Definitions

- **Co-occurrence Graph:[2]** A graph data structure that maintains a weighted set of co-occurrences of terms across the corpus. This graph approach is related to word clustering methods, where co-occurrences between words can be obtained on the basis of grammatical or collocational relations[5]. Formally we define the undirected co-occurrence graph G as

$$G = (T, P, w)$$

where, T is the set of all terms in the given corpus. P is the set of all pair-wise co-occurrence between terms in T. The function $w : P \rightarrow \mathbb{N}$ is the corresponding pair-wise co-occurrence count.

- **Continuous Learning:** Continuous Learning [3] is the process of constant improvement with no fixed end and the final goal is improvement itself. A continual learning algorithm is characterized by the following properties: (i) It should be autonomous. It must behave in its environment and be able to assign credits to desirable and undesirable behaviours. (ii) These behaviours should be capable of spanning for arbitrary periods of time i.e. the duration of a behaviour is not determined before hand. (iii) Continual Learning algorithms should acquire new behaviours only when useful.

2.2 Algorithm for Signature Generation for a class C

Before the start of algorithm, presence of co-occurrence graph, concept & containment hierarchy is presumed in the system. A class(or label) typically represents a "is-a" parent while instance represent all concepts that inherit that class. For example "Person" is a class while "Anoop" is an instance of "Person" class. Hierarchically 'Anoop' **is-a** 'Person'.

Algorithm for generating the signature is depicted in Algorithm 1

2.3 Algorithm for 'is-a' parent determination for an unknown concept

Prologue: This algorithm doesn't resolve the class of the unknown entity at an instant when a text segment is posted in a community website, rather it is resolved at at later stage subject to the condition given in Algorithm 2.

Algorithm for finding 'is-a' parent of an unknown entity is depicted in Algorithm 2

3. RESULTS

Dataset: Seekha (http://www.facebook.com/seekhain) is a academic networking portal. We used the dataset of 11K Seekha concepts and co-occurrence graph for our experiment. The dataset was already labelled.

Experiment and Results: From the co-occurrence graph, we created signatures for 20 main classes. The graph wasn't dense enough to create more signatures. Then we took 500

Algorithm 1 Algorithm for signature generation for a class

Find all instances of class C from the concepts hierarchy and store them in an array $'N'$.
From the co-occurrence graph find all the co-occurring concepts corresponding to each element of $'N'$ and store them in a 2-D associative array $'NC'$ such that key refers to a concept in $'N'$ and the value refers to an array which holds all co-occurring concepts corresponding to the key concept.
From the concept hierarchy we find the "is-a" parent of the co-occurring concepts in $'NC'$ and store it in a 2-D array $'SC'$ such that key refers to a concept in $'N'$ and the value refers to a multi-set array which holds "is-a" parent of all co-occurring concepts corresponding to the key concept.
Find the union of multi-set array $'SC'$ and find the signature as

$$\overrightarrow{S} = \sum_{k=1}^{X} j_k * CC_k$$

where CC_k is the concept class in the union of multiset $'SC'$, X being the total number of distinct classes CC_k and coefficient

$$j_k = \frac{total\ no\ of\ occurrences\ of\ CC_k}{total\ no\ of\ instances\ of\ C}$$

To narrow down the components of the signature $\overrightarrow{S}$ use 80-20 rule [1]. For applying this rule, order the components in descending order of their coefficient and select top u components CC_i for which the sum of co-efficient is 80% of the sum of co-efficient of all the 'n' elements.

$$\sum_{i=1}^{u} i = 0.8 * \sum_{i=1}^{n} i$$

Algorithm 2 Algorithm to find 'is-a' parent

Prologue: Presence of Co-occurrence Graph, Class Signature Database created using section 2.2 and Concept & Containment Hierarchy
Algorithm:
When a text segment is posted in a community website, extract concepts using a concept extraction algorithm and populate them in the co-occurrence graph.
Any new entity, not present in the knowledge base, is labelled as "Unknown" in the 'is-a' hierarchy.
Over time many of the unknown concepts in 'is-a' hierarchy are manually labelled randomly.
When an unknown entity has enough labelled neighbours (threshold can be heuristically set. For e.g. atleast 5 and 50% of all neighbours) in the co-occurrence graph, we create a candidate vector composed of 'is-a'(label) of the co-occurring neighbours of the unknown entity.
This candidate vector is compared against the signature database using cosine similarity to determine the most matched label.
Upon resolution of unknown entity to a class, signature of that class is updated using algorithm 1
If an unknown entity is resolved incorrectly, the threshold of labelled neighbours around an unknown entity is adjusted. The threshold for no. of instances and no. of labelled neighbours around the instances that participate in signature generation for a class, is also adjusted.

instances(concepts) assuming them as unknown that belonged to these classes, and determined their 'is-a' parent algorithmically. We were able to get 62% accuracy (percentage of concepts correctly labelled algorithmically) in class identification against the actual class for the instances.

4. CONCLUSION AND FUTURE WORK

We used the co-occurrence graph in a continuous learning environment to generate signature for a class (label). The signature database is later used to resolve unknown entities entered into the knowledge base. The signature for a class is updated when a new instance of that class is added to the system. We didn't use traditional supervised learning method since in a community setting, we didn't have training data in advance. The knowledge base evolves over time in community setting. Also the data churn in a community is tremendous. Introduction of new class can be easily accommodated in a continuous learning environment.

5. REFERENCES

[1] D. M. Powers. Applications and explanations of zipf's law. In *Proceedings of the joint conferences on new methods in language processing and computational natural language learning*, pages 151–160. Association for Computational Linguistics, 1998.

[2] A. R. Rachakonda, S. Srinivasa, S. Kulkarni, and M. Srinivasan. Mining analytic semantics from unstructured text. Technical report, Technical report, 2012.

[3] M. B. Ring. *Continual Learning in Reinforcement Environments*. PhD thesis, University of Texas at Austin, 1994.

[4] S. Srinivasa. Aggregating operational knowledge in community settings. In *On the Move to Meaningful Internet Systems: OTM 2012*, pages 789–796. Springer, 2012.

[5] D. Widdows and B. Dorow. A graph model for unsupervised lexical acquisition. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics, 2002.