Correlating Entities

R.V.Guha
Google
guha@guha.com

Abstract

An important step in automated knowledge base construction is the resolution of new entities against existing entities in the knowledge base. In this paper we describe a formal model for entity correlation and present some results on the conditions under which new entities can with high probability be correctly mapped to existing entities.

1 Introduction

References to things/entities (people, places, events, products, etc.) are ubiquitous. They occur in almost all communications, from natural language utterances to structured data feeds. Correctly resolving these references is vital to the proper functioning of many systems. Variations of this problem have been studied in fields ranging from philosophy and linguistics to database integration and artificial intelligence.

This paper is an adaptation of the general treatment in [5] to the entity reference problem in the context of automatically constructing knowledge bases.

2 Problem Statement

During the Automated Construction of Knowledge Bases (AKBC), we encounter references to entities along with various facts about these entities. These may be new entities that the KB does not yet know about or they could be references to entities already in the knowledge base. An important step in AKBC is to correctly reconcile these references.

In most sources used in AKBC, such as news articles, references to people, places, organizations, etc. are usually accompanied by descriptions. As humans, we use these descriptions to uniquely identify and correlate these entity references to known entities. For example, imagine we come across a reference to a person called Michael Jones. Given the number of people with that name, the name alone is highly ambiguous. However, if we augment the name with the person's date of birth, his profession, etc., this description fairly quickly uniquely identifies the person.

During AKBC, we would like to automatically use these descriptions to correctly resolve entity references. There are four important variables governing the effective use of descriptions to resolve references. They are:

1. Description Size: Given a description for an entity, we would like to know whether it uniquely identifies the entity. Going back to our example, if the description was simply 'a person named Michael Jones', even if the KB had only a single Michael Jones, it would likely be incorrect to assume that the new entity and the existing entity in the KB are the same. How many more facts about this person do we need to add to the description before we can be sure that this new entity is the same as (or different from) the one in the KB?

To determine this precisely, we need the probability distribution of the different possible descriptions. Given the difficulty of obtaining this distribution, how can we approximate it? Under such an approximation, what are the average, lower and upper bounds on the required description? In Information Theoretic terms, what is the minimum number of bits of information in a uniquely identifying description?

2. Shared Knowledge: Shared knowledge about the world is key to interpreting descriptions. Consider a description such as 'a person named Michael Jordan, professor in the CS department at UC Berkeley'. If the KB also knew of the Michael Jordan who is a professor in the CS department at UC Berkeley, this description should be enough to uniquely identify the person. On the other hand, if the KB did not know where its Michael Jordan worked, or had him still listed as teaching at MIT, this description would not be adequate. The description would have to contain other pieces of information about him in order to correctly identify him. How do we measure how much knowledge is shared?

3. Shared language: Symbols whose meaning is shared are required for constructing descriptions. Consider a (part of a) description 'a person who lives in X', where X could be something unambiguous (e.g., Canada) or something ambiguous (e.g, Madison). The former is more useful for resolving the entity reference. Of course, 'lives in' itself is assumed to unambiguously map to a KB relation. What is the minimum number of unambiguous terms that need to be shared between the source and the KB so that with high probability, every ambiguous term (i.e., new entity reference) can be correctly disambiguated?

4. Structure of the domain: Finally, the structure of the underlying world also plays a big role in descriptions. Consider two groups of people, one in which there are many different names, employers, interests, etc. and another in which there are far fewer distinct names, only a couple of employers, etc. In the first case, the richness of structure in the underlying world make it easier to construct descriptions that pick out individuals. How do we measure the richness of the structure of the underlying domain?

We are interested in the relation between these four variables. After a brief review of related work, we discuss formal models of the knowledge base (that we are trying to construct) and of descriptions. When then use it to quantify the relation between the four variables listed above.

3 Related Work

The problem of correlating references to entities across systems arises in many different fields, including statistics, epidemiology, history, census analysis, database integration, privacy protection, linguistics and communication. The problem goes under many different names, including "record linkage", "list washing", "merge/purge processing", "data matching", "entity disambiguation", "coreference resolution" and "database hardening".

All of these cases are solving a problem very similar to the one we face with AKBC. Given a new entity, how do we determine if/which existing entity it maps to? In each of these cases, we construct a description of the entity, hopefully one that uniquely identifies it and match it to those of existing entities.

The research by ([4], [2] and [7]) are archetypal of the approaches that have been followed for solving this class of problems. Much of the attention has focussed on the development of algorithms capable of correctly performing the matching between simple descriptions of entities. Further, most of the work has focussed on overcoming the lexical heterogeneity of the representation of the string values and on differences introduced by data acquisition and entry errors.

The work presented here differs in two main respects. Firstly, the data/representation model used to encode information about each entity is more expressive, allowing for arbitrary relational information. The methods proposed in the research on data integration typically do not extend to complex relational structures. Secondly, our goal is not to come up with a specific matching algorithm, but to establish a general framework and derive bounds on the knowledge that must be shared and for the minimum length/information content of the description for the matching to be possible at all.

4 Model of KB

Let the KB that we are trying to build be represented as a directed labelled graph. We will henceforth refer to the KB as the graph G and the adjacency matrix of the graph as M. Let the expected number of entities in the KB (i.e., the number of nodes in G) be N. Of course, during most of the construction, the KB will have fewer than N entities.

We first need a mathematical model for our graph. We assume that our graph is created by a stochastic process. There has been extensive work on modeling graphs created by stochastic processes, most of which can be easily extended to labelled graphs. We begin with a set of N vertices and then add edges between pairs of vertices according to some probability distribution. Different probability distributions give us graphs with different kinds of properties. The most studied is the Erdos Renyi model, denoted $G(N, p)$, in which we have a graph with N nodes and every possible edge occurs independently with probability p. In the labelled graph variant of this model, we have a probability distribution where the the probability of the arc between any pair nodes having the label L_i is p_i, with the absence of any arc being considered a special arc which we shall refer to as L_{null}.

Many other models have been proposed for random graphs. Recently here has been considerable work on other random graph models [6], such as those involving preferential attachment, which can be useful for modelling structures such as the web. Some systems use more 'regular' graphs (a grid being an extreme example of such a regular graph). Database systems with strict schemas are a good example of this. The choice of graph model depends on the details of the underlying world that we are trying to construct a knowledge base about. The analysis presented in this paper can be used with any of these models, so long as the following assumption/approximation holds.

Ergodicity Assumption: We make the assumption/approximation that some minimum number of rows in the adjacency matrix are generated by an ergodic process. This basically means that different randomly chosen long enough substrings from these rows in the adjacency matrix should have the same distribution of arc labels. More concretely, randomly chosen long enough samples from these rows in the adjacency matrix should obey the asymptotic equipartition property (AEP) [3]. The AEP states that if we have a process generating strings of length K according to a probability distribution that has an entropy H, the set of 2^K possible strings can be partitioned into two sets: the first set of size 2^{HK}, which is called the typical set, of strings that are likely to occur, and the second set, containing the remaining strings, that are not likely to occur. Each of the strings in the typical set are have an equal probability of occuring, which is 2^{-HK}. H, the entropy of strings in the adjacency matrix is the quantitative measure of the richness of the structure of the underlying domain.

This approximation allows us to make uniform estimations about the size and information content of uniquely identifying descriptions.

When the source and KB have different views of the world, we construct the adjacency matrix for each of their views of the world and use the mutual information M between these two matrices to quantify the shared knowledge.

5 Descriptions

A description of a node is any subgraph of the graph, which includes that node and some (possibly none) of the nodes whose names are shared. Since any subgraph that includes a node is a description of that node, every node will have many descriptions. Some of these descriptions may uniquely identify the node.

Descriptions come in many different 'shapes'. The computational complexity of dereferencing a description is a function of its shape. If a description is an arbitrary subgraph, dereferencing it involves solving a subgraph isomorphism problem, which is known to be NP-complete. However, if we impose some restrictions on the structure of admissible descriptions, the complexity of decoding the description can be kept down. In this section, we look at a few different kinds of descriptions with different levels of decoding complexity.

Assume that K of the nodes $< S_1, S_2, ...S_K >$ are unambiguous. We have M arc labels: $< L_1, L_2, ...L_m >$. Given a node X (a reference to which might be ambiguous), we need to construct a description for this node. Let the relation between this node and the i^{th} of the K nodes be L_{xi}.

The relation could be a direct arc between the two nodes or a more complex path. The simplest class of descriptions, which we will refer to as 'flat descriptions', corresponds to the logical formula:

$$L_{x1}(X, S_1) \wedge L_{x2}(X, S_2) \wedge ... \wedge L_{xK}(X, S_K)$$

In this class of descriptions, if there is no direct arc between X and the shared node S_i, we use the special arc label L_{null}. This class of descriptions can be decoded very efficiently, using standard database techniques.

We can also write this as the string $L_{x1}L_{x2}L_{x3}...L_{xK}$. If the columns corresponding to the K nodes whose names are shared are placed adjacent to each other in the adjacency matrix, this string is simply the entries in those columns for the row corresponding to X in the adjacency matrix. As mentioned earlier, the only assumption we make about the graph is that these description strings (i.e., the rows/columns of the adjacency matrix corresponding to the K shared terms) obey the AEP. The entropy of this class of description strings is:

$$H_d = -\Sigma p_i log(p_i)$$

where p_i is the probability of the label L_i occuring between two randomly chosen nodes in the graph. Note that H_d, the entropy for this simple class of descriptions is the same as the entropy of strings in the adjacency matrix, i.e., For simple descriptions, the entropy of the descriptions is equal to the entropy of strings in the adjancency matrix, which is also the measure of the richness of the structure of the domain. Similarly, when the source and KB have different views of the world, M_D, the mutual information between descriptions as construed by the source and KB, is equal to M, the mutual information between the adjacency matrix as seen by the source and KB.

More complex descriptions emerge when, instead of using L_{null} for the case where there is no direct arc between X and S_i, we allow paths or of length longer than 1. More generally, we can allow arbitrary intermediate subgraphs connecting X and S_i, involving multiple intermediate nodes with arcs between these intermediate nodes. Depending on the class of intermediate subgraphs allowed, we get different kinds of descriptions with different levels of dereferencing complexity. In increasing order of complexity, we can restrict ourselves to strict paths, trees, planar intermediate subgraphs or allow for arbitrary intermediate subgraphs. As the complexity of the allowed intermediate graph increases, the number of possibles shapes for the graph and hence the entropy of the descriptions increases.

In this paper, we restrict our analysis to descriptions where the intermediate graph is of some fixed size D. Let us name the set of possible graphs of size D with an arc label set $< L_1, L_2, ...L_m >$ as $< L_{null}, L_{D1}, L_{D2}, ... >$. If $D = 1$, then this set is just $< L_{null}, L_1, L_2, ...L_m >$. When $D > 1$, the description for X looks the same as when $D > 1$, except, when there is no direct arc between X and S_i, we check to see if there is an intermediate graph of size $\leq D$ connecting X and S_i and if there is, we use the corresponding name for it. Let the entropy of this description string be H_D. Consider a transformation of the adjacency matrix where the L_{null}s are replaced with the appropriate terms from $< L_{null}, L_{D1}, L_{D2}, ... >$. We call this the $DescriptionAdjacencyMatrix$. H_D is the entropy of strings from this adjacency matrix and M_D is the mutual information between the sender's and receiver's views of this adjacency matrix.

H_D, the entropy of the adjacency matrix of descriptions is the measure of the richness of the structure of the underlying world. M_D, the mutual information between the views of this matrix from the perspective of the source and KB is the measure of the shared knowledge.

6 Results

We now discuss the relation between the four variables governing the effective use of descriptions to resolve entity references:

1. Description size
2. Shared Language, as measured by the number of symbols whose meaning is apriori shared

3. Structure of the underlying world, as measured by the entropy of the adjacency matrix of descriptions
4. Shared Knowledge, as measured by the mutual information between the views of this matrix from the perspective of the source and KB

The following theorems (proofs are given in [5]) captures the relationship between these variables.

Theorem: Let there be $C_s log(N)/H_D$ unambiguous nodes used to construct descriptions, where H_D is the entropy of the descriptions and N is the number of nodes in the graph. For large graphs, if $C_s \geq 2$ then, with high probability, we can correctly disambiguate references to all but a constant number of the other nodes. If $C_s < 2$, then, with high probability, there will be more than a constant number of nodes that cannot be disambiguated. The description for each node is a string of length $C_s log(N)/H_D$ that contains the relation (from the Description Adjacency Matrix) between that node and each of the nodes with shared names.

Theorem: In the case where there is a difference in the view of the adjacency matrix as viewed by the source and KB, the number of unambiguous terms required is $2log(N)/M_D$ where M_D is the the mutual information between the views of the graph that the source and KB. The description string is computed as before.

From the above theorems, it follows that we need at least $2log(N)/H_D$ (or $2log(N)/H_D$) shared symbols. When there is a difference in view of the underlying world, between the source and the knowledgebase, i.e., shared knowledge reduces, the number of terms that need to be shared increases.

Each description (on the average) is a string of length $2log(N)/H_D$ and entropy H_D. The information content of the description is hence $2log(N)$. It is very interesting to note that the information content of the description is independent of the entropy of the underlying graph. So, if the information content of the all the information about the entity from a source is less than this, we will have an ambiguity.

For the case where there is a difference in the view of the world between the source and the KB, the description has to be of length $2log(N)/M_D$ and the information content of the description has to be at least $2log(N)H_D/M_D$. H_D/M_D increases as the view of the world between the source and KB diverges, i.e., shared knowledge decreases. As this happens, the size of the required description increases.

7 Conclusion

In this paper, we introduced a formal model of entity reconciliation and presented some results on certain conditions that must be met for successful entity resolution.

Our next step is to use this model for entity resolution during the automated construction of knowledge bases. More specifically, we would like to determine when two entity references, one from the source and one from the KB under construction, refer to the same entity. The source, often relatively small, like a web page, contains a number of facts about each entity. We can consider all these facts together as the description of the entity from the source. We would like to determine whether this set of facts can uniquely identify that entity. Further, if there is some entity in the KB that matches some subset of the description, we would like to determine whether that subset is enough to uniquely identify that entity. To do this, we need to first compute the entropy of the structure of the underlying world.

We are beginning some experiments, using this approach, for adding facts from web pages (encoded using Schema.org markup) to Freebase [1]. We hope to report results on this soon.

References

[1] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *2008 ACM SIGMOD*, pages 1247–1250. ACM, 2008.

[2] W. W. Cohen, H. Kautz, and D. McAllester. Hardening soft information sources. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 255–259, New York, NY, USA, 2000. ACM.

[3] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

[4] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19:1–16, 2007.

[5] R. V. Guha. Communicating and resolving entity references. arxiv.org/abs/1406.6973, 2014.

[6] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[7] H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser. Identity uncertainty and citation matching. In *In NIPS*. MIT Press, 2003.