CS 410: Text Information Systems

Technology Review of Google's Knowledge Vault

Name: Liu, Shi Yao
UIN: 676421467

Introduction

Knowledge bases are typically built on information from direct contributions by human volunteers as well as well-structured repositories. This technique limits knowledge bases in multiple ways. Frequently mentioned properties of frequently mentioned entities dominate the entries. Wikipedia inboxes are often used as sources of information. However, Wikipedia growth has been observed to have plateaued, hence also impacting the growth of knowledge bases. Researchers at Google proposed a new approach to construct a probabilistic knowledge base (known as "Knowledge Vault" or "KV" in short) to overcome these limitations. In this system, information is automatically extract from the entire web. However most of such information is unstructured, noisy, and contains unreliable facts. In order to realize this new system, the researchers have also developed a way to evaluate the truthfulness of the extracted information.

Each piece of knowledge information is represented by an RDF triple which is a triplet of values representing a subject, predicate, and object. For example: </m/02mjmr, /people/person/place_of_birth /m/02hrh0_> where "/m/02mjmr" is the ID for Barack Obama, and "/m/02hrh0_" is the ID for Honolulu. This representation separates the facts from their lexical representation to remove redundancy that arise from different wordings of the same information as well as make knowledge language-independent. Associated with each triple is a confidence score which represents the probability the Knowledge Vault believes the triplet is correct.

Scope

The scope of this technology review covers the important aspects of the Knowledge Vault in terms of its functionality, methods used to overcome challenges, summary of the effectiveness of the methods used, additional considerations, and future work.

In addition to the review the Knowledge Vault and the work done by the researchers, this technology review proposes possible applications of the system and the techniques used to construct it as an extension. The methods developed by the researchers to construct the Knowledge Vault are very general and powerful. Thus they can be applied to fields and applications requiring highly specialized domain-specific knowledge as well as simpler implementations where problems encountered by the researchers are not applicable. Knowledge Vault is meant to be a repository of all human knowledge. For application-specific use cases, the problem statement becomes simpler since the entities and relationships between them are restricted, and the sources of information may also be less arbitrary.

The methods used to construct Knowledge Vault is covered in the sections "Labeling Data", "Information Extraction", "Generating the Prior", and "Extractor and Prior Fusion". The most notable components of the Knowledge Vault are the the prior model, information fusion, and the local closed-world assumption (LCWA) used for training. The effectiveness of these three components are summarized in the "Results" section. Future work, possible improvements and thoughts from the researchers are covered in the section "Additional Considerations". Finally, a section not covering the work of the researchers is "Possible Applications". This section is an

extension, it explores possible applications of Knowledge Vault and the techniques used in its development.

Labeling Data

All the components of the system use supervised machine learning methods to fit binary classifiers. The classifiers determine the probability of a given RDF triple being true. To determine the labels, an assumption called local closed-world assumption is made. For triples that exist in Freebase, assume the label is "true". For triples that do not exist in Freebase, assuming the label is "false" would be too naive because Freebase is far from complete. Let (s, p, o) represent a subject, predicate, object triple and O(s, p) represent the set of object values for a given s and p. If (s, p, o) exists in O(s, p), then the label would be true. If (s, p, o) does not exist in O(s, p) but O(s, p) is not empty, then (s, p, o) is labelled false. Finally, if (s, p, o) is not in O(s, p) and O(s, p) is empty, (s, p, o) is discarded. The idea here is the knowledge base is assumed to be locally complete for the subject-predicate pair. If O(s, p) is empty, nothing can be said about the triplet (s, p, o) so it should be discarded rather than labelling it false. This method does have drawbacks since the knowledge base in a lot of cases does not contain all the objects that should be in O(s, p) but the researchers did still demonstrate it to be effective.

Information Extraction

The source of prior data is Freebase, a knowledge base of structured data containing high-confidence facts. Of course this knowledge base is far from complete, but it is well-suited as a starting point and as a basis to evaluate the truthfulness of new information. Extractors for four types of sources of web data were made to extract knowledge: free text, HTML DOM trees, HTML web tables, and human annotations of webpages. To prevent over counting of evidence, each triple is counted once per domain as opposed to per URL.

To extract information from free text, NLP is is heavily used for information extraction. For each predicate, a set of seed entity pairs with this predicate is chosen from the existing knowledge base and more examples of sentences with these patterns occurring between these entities is chosen. The local closed-world assumption is used to derive labels for the extractions, and a binary classifier is trained for each predicate using MapReduce. The score of the extracted triples is the classifier's output.

To extract information from DOM trees, classifiers are trained as in the free-text case, with one difference. Instead of extracting entity relations from text, features connecting two entities from the DOM trees are used. Specifically, the lexical path along the tree between two entities is used as a feature vector that describes the relationship between the two entities. The score of the extracted triples is the classifier's output.

For tables, the columns represent the relationship between the entities defined in the table. Reasoning about which predicate the columns represent involves trying to find a match in Freebase. Columns marked as ambiguous are discarded. The score of the extracted triple reflects the confidence given by the entity linkage system.

Many human annotated pages are related to events or products. Such information are not currently stored in KV. A small set of fourteen predicates mostly relating to people are used. A manual mapping between schema.org and Freebase is defined for the se predicates. The score of the extracted triple reflects the confidence given by the entity linkage system.

Generating the Prior

Reliable prior knowledge can assist in combating the noisy and unreliable nature of information extracted from the web. Existing triples in Freebase is used to fit the prior models. Given any triple, a probability of the information being true can always be determined even if no corresponding evidence exists on the web. This can be thought of as link prediction in a graph (predicates connecting entities, and how likely is a particular predicate to exist between two entities).

One approach to solve this problem is to use the path ranking algorithm (PRA). Start with a set of pairs of entities that are connected by some predicate p. PRA performs a random walk on the graph, starting at all the subject nodes. Paths that reach the object nodes are considered successful. Since multiple rules or paths might apply for any given pair of entities, they can be combined by fitting a binary classifier. The features are the probabilities of reaching O from S following different paths, and the labels are derived using the local closed-world assumption. For testing a new triple, look up all the paths for predicate p chosen by the learned model, and perform a walk on the training graph from the subject to the object via each such path. This gives a feature value that can be plugged in to the classifier.

Another approach is to view link prediction as tensor completion in a sparse E x P x E matrix G, where E is the number of entities (subjects and objects) and P is the number of predicates. G(s, p, o) = 1 if there is a link of type p from s to o. Otherwise, G(s, p, o) = 0. A multi-layer perceptron network with 60 hidden layers is trained for the model:

$$\Pr(G(s,p,o)=1) = \sigma\left(\vec{\beta}^T f\left[\vec{A}\,[\vec{u}_s, \vec{w}_p, \vec{v}_o]\right]\right)$$

$f()$ is a non-linear function such as tanh. $\vec{A}$ is a L × (3K) matrix matrix representing the first layer weights (the 3K term arises from the K-dimensional u_s and w_p and v_o). $\vec{\beta}$ is a L × 1 vector representing the second layer weights where L = K = 60. To demonstrate that the neural network learns a meaningful semantic representation of the entities and predicates, compute the nearest neighbors of various items in the a K-dimensional space. From previous work, it is known elated entities cluster together, so the focus here is on predicates. The model indeed learns to put semantically related but not necessarily similar predicates close together.

Extractor and Prior Fusion

To combine the signals from the four extractors, construct a feature vector f(t) for each of the extracted triple t = (s, p, o) and apply a binary classifier to compute Pr(t = 1 | f(t)). The feature

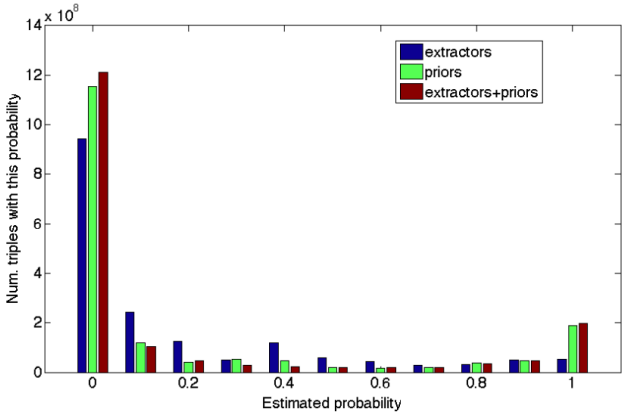
vector consists of the square root of the number of sources and the mean score of the extractions from the extractor (averaging over sources). The motivation for using square root is to reduce the effect of very commonly expressed facts. Similarly, log(n+1) can be used, where n is the number of sources. Note that duplicate sources are removed before this operation (recall each triple is counted once per domain as opposed to per URL).

Priors can be fused similarly, except the only difference is the feature used. Since there are no extractions, the feature vector contains the vector of confidence values from each prior system as well as indicator values specifying if the prior was able to predict them or not (to distinguish a missing prediction and a score of zero).

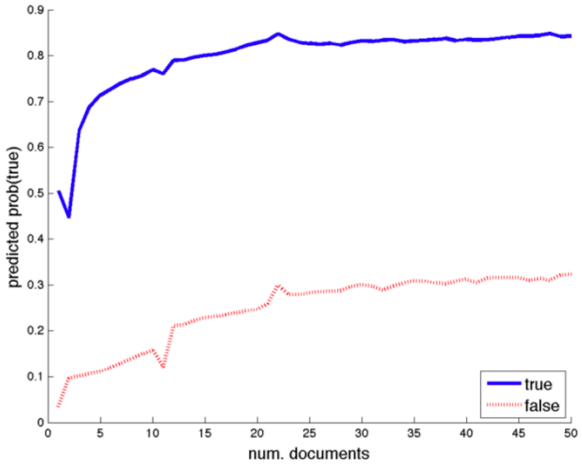
Results

Two methods were used to generate the graph-based prior model, one using PRA and one using a neural network. Surprisingly, both methods performed similarly when comparing ROC curves (the AUC scores).

The Knowledge Vault introduces information fusion. Specifically, the fusion of extractors and priors. Combining priors and extractors increases the number of high-confidence triples. In other words, the certainty of true triples increases and the number of triples that can be clearly labelled as true also increases.



In addition, increasing the number of sources also increases the confidence of true triples as illustrated by the figure below. Note that each source is unique on domain as opposed to URL.



It is well-know that features play a significant role in the performance of a model. Since the prior plays such a major role in this system, it is important to examine the validity of LCWA used to label the training data to train the models. Recall LCWA is the assumption that the knowledge base containing prior knowledge is locally complete. In other words, it is the idea that Freebase only contains approximate knowledge (since it is incomplete). One example to illustrate this problem is Freebase's records for actors for a movie. Even if a movie has 20 actors, Freebase will only list the top five. Therefore, if an actor is presented to the system and an entry does not exist, it is unreasonable to make the assumption that the information is false. This is the basis for LCWA used for training models used in KV. In order to evaluate the validity of this assumption, the researchers manually labelled a subset of their balanced test set using a team of in-house raters. The set contains 1000 triples for 10 different predicates. Raters label the triples either as true, false, or unknown based on their own research. The triples labelled as unknown (305 in total) are discarded. The researchers then computed the performance of the system on this labelled test set against both LWCA and human labels where in both cases the system is trained on the full training set labelled with LWCA. The result is that the performance of the human label is only slightly lower, which indicates that LWCA assumption is indeed valid. The AUC scores are summarized as follows:

Labels	Prior	Extractor	Prior+ex
LCWA	0.943	0.872	0.959
Human	0.843	0.852	0.869

Additional Considerations

For reasons of scalability, each fact is treated as an independent random variable that is either true or false. In reality, many triples are related. This limitation can be illustrated with the following example: for a relation such as "born in", there can only be one true value, so the (s, p, o_i) triples representing different values o_i for the same subject s and predicate p are correlated because of mutual exclusion. A simple way proposed by the researchers to handle this is to collect together all candidate values and force the distribution over them to sum to 1, possibly also allocating some extra probability mass to account for the fact that the true value might not be among the extracted candidates. Preliminary results show that this method is unreliable, so the researchers are working on more sophisticated methods.

Soft constraints are not accounted for. An example is the fact that people typically have 0 to 5 children but this distribution has a long tail since there does exist people with more than 5. However, it would still be surprising if we extracted 100 different children for one person, even though the probability exists (this may even indicate an error). Preliminary experiments using joint Gaussian models to represent correlations amongst numerical values show some promise. The researchers are still working on integrating this kind of joint prior into Knowledge Vault.

Values can also be represented at multiple levels of abstractions. Information can be represented at different degrees of granularity. For example, Obama is born in Honolulu. The

fact that Obama is born in Hawaii is also true. Knowledge Vault uses geographical knowledge to generalize cases like this, but researchers are also working on ways to generalize this technique.

Recall that belief in a triple increases as it is extracted from more sources. This is a problem if sources are correlated or duplicated even. The current simple solution to combat this is to count each domain only once. However, more sophisticated copy-detection mechanisms are in the works.

The truth of a fact may change over time, such as the CEO of a company. Freebase allows facts to be annotated with beginning and end dates. There are plans to extend KV to temporal information. Although it seems simple, it is actually non-trivial since the duration of a fact may not be related to the timestamp of the source.

As the world changes, new content is also added to the web, and the number of entities also increases. Therefore, there needs to be a way to automatically create new entities inside KV. The addition of new entities must be cone in a controlled way, since redundant and synonymous representations and relations may exist.

The RDF triple is a simple representation for information and seems adequate for factual information. However, consider the difference between running and jogging or the difference between jazz and the blues. In a lot of cases, there is a long chain of concepts that are difficult to represent with a simple fixed ontology. Neural networks may be able to accomplish such a task, but this is left to future work.

Finally, overcoming all technical challenges still leaves a fundamental upper bound. The vision for Knowledge Vault is for it to be a complete repository of human knowledge. Even if a perfect reading system was created, not all human knowledge is available on the web. Common sense knowledge is difficult to acquire from text sources. Crowdsourcing techniques may be able to overcome this information barrier.

Possible Applications

One direct application of Knowledge Vault is screening for false information. For example, social media platforms came under heavy scrutiny recently for propagating false information on COVID19 and vaccines. Health-related information posted are unchecked and given the large volume of information, screening every single one would be very difficult. A system similar to Knowledge Vault can be constructed using domain-specific information (vaccine, virology, etc) and such a system would be able to evaluate the validity of shared information. As more guidelines change or becomes available from health organizations, more vaccines become available, and new variants emerge as a result of mutations, the system can add new information to its knowledge base, thus keeping up with developments.

Another application is assisting in the design and development of hardware. Using iPhone as an example, each iteration becomes more and more complicated as more components are added. It is widely known that various hardware components interfere with each other, especially in a small form factor such as an iPhone. By crawling the internal bug reporting system, factory test database, and customer reporting system, it would be possible to find

relations between various components. Using the RDF triple format, a possible entry could be: "display" (subject), "causes high frequency signals on" (predicate), "data bus" (object), and another entry could be: "data bus" (subject), "causes RF harmonics on" (predicate), "receiver chain" (object). Hardware components are entries, and the interference or relationship between them are predicates. A knowledge vault constructed in this way can assist designers and testers predict issues and assist debug by querying the knowledge base to identify problematic components and most probable causes. As new bugs are reported and more factory data becomes available, information can be evaluated and added. The local closed-world assumption used for Google's Knowledge Vault is especially important for collecting user reports. Out of the three major sources of information, customer reports are the least reliable (most reliable being factory test data, followed by internal engineering bug reports). Therefore, it is very important to evaluate the confidence of customer-provided information reliably.

References

This report covers information and uses figures from the following paper:

X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, W. Zhang. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. *In The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 601-610, 2014.