LONLIES: Estimating Property Values for Long Tail Entities

Mina Farid
University of Waterloo
mina.farid@uwaterloo.ca

Ihab F. Ilyas
University of Waterloo
ilyas@uwaterloo.ca

Steven Euijong Whang
Google Research
swhang@google.com

Cong Yu
Google Research
congyu@google.com

ABSTRACT

Web search engines often retrieve answers for queries about popular entities from a growing knowledge base that is populated by a continuous information extraction process. However, less popular entities are not frequently mentioned on the web and are generally interesting to fewer users; these entities reside on the long tail of information. Traditional knowledge base construction techniques that rely on the high frequency of entity mentions to extract accurate facts about these mentions have little success with entities that have low textual support.

We present LONLIES, a system for estimating property values of long tail entities by leveraging their relationships to head topics and entities. We demonstrate (1) how LONLIES builds communities of entities that are relevant to a long tail entity utilizing a text corpus and a knowledge base; (2) how LONLIES determines which communities to use in the estimation process; (3) how we aggregate estimates from community entities to produce final estimates, and (4) how users interact with LONLIES to provide feedback to improve the final estimation results.

1. INTRODUCTION

Search engines can provide direct answers to queries about popular entities, e.g., capitals of countries and birth dates of celebrities. These answers are obtained from structured knowledge bases that contain information about entities that are frequently queried and are generally interesting to a large number of users. Answering a user query becomes a matter of understanding and representing the user's intent in an meaningful way, then finding the corresponding nodes and edges in the knowledge base graph that contain the answer.

While some knowledge bases extend their content by allowing users to manually add information, many recent approaches [4, 7] have focused on automating the extraction of facts from unstructured data such as text. Most fact extraction techniques rely on the abundance of text and utilize the redundancy of information that is extracted from multiple data sources in order to statistically validate the newly extracted facts. For example, Google's Knowledge Vault [4] extracts fact triples using four different types of extraction systems, and the confidence of a triple increases as more extractors produce the same triple and more unique web sources contain the triple. The knowledge base is then augmented with facts that are extracted with high confidence, growing its size to cover a larger domain of information.

SIGIR '16, July 17-21, 2016, Pisa, Italy
© 2016 ACM. ISBN 978-1-4503-4069-4/16/07...$15.00
DOI: http://dx.doi.org/10.1145/2911451.2911466

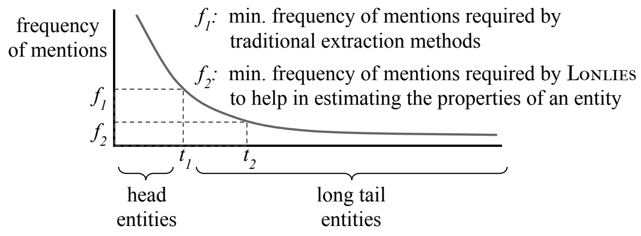


Figure 1: Mentions of Head and Long Tail Entities

Figure 1 models the frequency of entity mentions in the text as a distribution, representing the entities on the x-axis and the frequency of mentions of these entities on the y-axis. The "head" entities have many mentions (at least f_1 in the figure) that allow for efficient extraction using traditional methods. Entities that reside on the long tail (beyond the point t_1) pose a challenge to automatic knowledge base construction methods due to the insufficiency of relevant web content; the extracted facts cannot be verified and questions about them cannot be directly answered.

LONLIES estimates property values for long tail entities and does not rely on direct extraction of these values from text. LONLIES utilizes fewer mentions of these entities and information from a knowledge base to produce probabilistic estimates for the property values. In Figure 1, LONLIES can extract properties of long tail entities (between t_1 and t_2) that have enough frequency of mentions to assess their relationships to head topics, but do not have enough redundancy to allow for efficient traditional fact extraction. The intuition behind LONLIES is to leverage the text corpus to find occurrences of the long tail entity e_l with other head entities with known properties and exploit those head entities as an "information context" around e_l, which we call communities, to estimate a value for a property p of e_l. Communities are perceived as potential populations that e_l might be a member of, and they provide a more general context about e_l. The main insight behind LONLIES depends on two observations:

- Judging if e_l belongs to a particular community c_i requires less mentions in the relevant text than extracting an exact value for p.
- The ability to find an estimation for a property value p from a group of entities that form a community c_i depends on the p values of the entities in c_i and other statistical properties of c_i (e.g., its size and the variance of p values) but does not depend on e_l.

We combine these two notions in the following estimator for $e_l.p$:

$$e_l.\hat{p} = f_{\forall c_i \in C}\big(M(e_l, c_i), c_i.\hat{p}\big)$$

where $M(e_l, c_i)$ measures the likelihood of e_l being a member of a community c_i, and $c_i.\hat{p}$ is an estimator for the property p that produces possible values for p using the p values of entities in c_i. $c_i.\hat{p}$ associates each estimate with a confidence interval that reflects the quality of the estimation. f is an aggregation function that combines estimates from a set of communities C.

The ability of Lonlies to find correct estimates for $e_l.p$ depends on (1) the estimation power of the constructed communities (i.e., how accurately $c_i.\hat{p}$ represents the values of p for entities in c_i); and (2) the confidence that e_l belongs to the communities used in the estimation. These two constraints together verify that estimates are produced from relevant communities that confidently produce representative p values. The error in estimating $e_l.p$ is the compound error of the membership assessment function M and the error in computing $c_i.\hat{p}$ estimates from communities.

Lonlies alleviates the risk of low-quality estimates for $e_l.p$ by being conservative in assessing the membership M and when producing estimates from c_i as we explain in Sections 2.3 and 2.4. Lonlies produces estimates only for results with high confidence. This, of course, lowers the recall of the long tail entities that Lonlies can handle, but maintains a high precision for the entities that are answered.

While this formula estimates $e_l.p$ from the p values of the constructed communities, the technique can be easily extended to leverage external rules and correlations of properties to estimate one property from another (e.g., to estimate *age* from *birth date* or *nationality* from *country_of_birth*).

2. THE Lonlies SYSTEM

Figure 3 illustrates the system architecture of Lonlies. We give an overview about the functionality of the four modules that constitute the stages of the estimation process and then discuss the details of each module separately. The modules rely on the availability of a text corpus and a knowledge base. The input to Lonlies is a long tail query entity, e_l, and a target property to estimate, p.

- The *KB Anchoring* module retrieves a set of documents that are relevant to the long tail query entity from the text corpus. It also extracts head entities that exist in the knowledge base that appear in the retrieved documents (Section 2.1).
- The *Community Construction* module builds a set of communities that may contain e_l as a member if e_l is added to the knowledge base (Section 2.2).
- The *Membership Assessment* module measures the similarity between e_l and a particular community in order to filter out irrelevant communities. It also filters out communities with low confidence of estimation (Section 2.3).
- The *Estimation* module aggregates estimates that are generated from the relevant communities and produces a distribution of estimates for the target property p (Section 2.4).

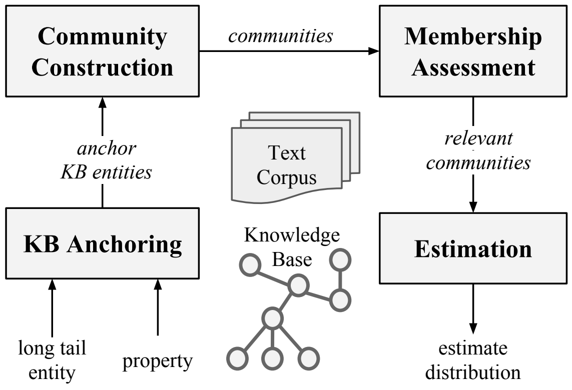


Figure 3: Lonlies System Architecture

2.1 Anchoring to Knowledge Base

The first step of Lonlies is retrieving a set of text documents that are relevant to the long tail query entity e_l. Lonlies extracts head entities that appear in the retrieved documents and exist in a knowledge base.

The co-occurrence of the long tail entity with head entities in text documents implies relationships to these entities in the knowledge base; the co-mentions represent what nodes in the knowledge base graph e_l might have been linked to if it was added to the knowledge base. Hence, we refer to these head entities as anchor KB entities E_H.

Lonlies does not classify the relationship between the long tail entity and anchor head entities that appear with it in the text. In fact, the co-occurrence merely indicates that a relationship between them exists. The *Membership Assessment* module validates these potential relationships.

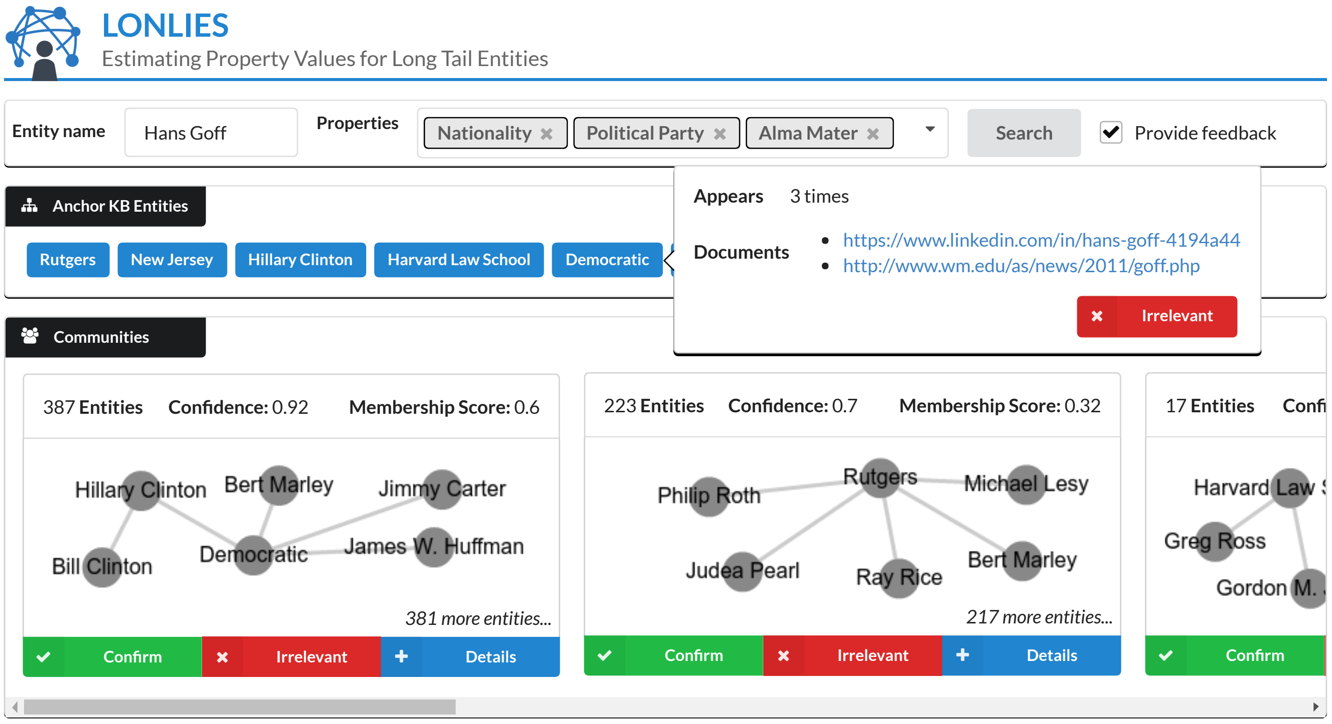
2.2 Community Construction

Given the list of head entities that the long tail query entity is mentioned with, E_H, the goal of this module is to construct a set of communities C that can produce high-quality estimates for p.

Definition 2.1. A community c consists of a set of head entities that are retrieved from a knowledge base. Entities of a community c reflect nodes in the knowledge base graph.

Communities are constructed by traversing the knowledge base graph starting from E_H entity nodes. In Lonlies, we implemented two methods to construct communities.

Coherent community detection approach. We materialize a k-hop graph around all E_H entities from the knowledge base. We then run a community detection algorithm similar to [6] to detect communities in the materialized graph. We modified the *betweenness* function of the algorithm to separate communities that have coherent p values instead of dense connected subgroups. This method produces communities with consistent p values.


Entity name: Hans Goff

Properties: Nationality, Political Party, Alma Mater

Provide feedback: ✓

Appears 3 times

Documents
- https://www.linkedin.com/in/hans-goff-4194a44
- http://www.wm.edu/as/news/2011/goff.php

Figure 2: Lonlies GUI showing Anchor Head Entities and the Constructed Communities

Common KB relation approach. In this method, we start from each entity in E_H and construct a community from all entities in the KB graph that share a particular relationship to it. For example, a community c may include all entities that *work_at* the *University of Waterloo*. Not all constructed communities can produce high-quality, useful estimates, and the estimation power of a community depends on the property p that we are estimating. For example, a community that consists of people *born_in* the *United States* can confidently estimate the *nationality*, but the same community cannot help in estimating *age* or *occupation*.

After constructing a possibly large number of communities, LONLIES calculates multiple statistical features for each community, including the confidence of its produced estimates which depends on the size of the community and the homogeneity of the p values of its members, depending on the type of p. LONLIES computes the variance of the p values if p is a numerical property, or Simpson's diversity index[1] [8] of the p values for categorical properties. LONLIES associates a confidence score with the produced estimates and discards communities with low estimation confidence. This drastically reduces the number of communities to consider in the following stage.

2.3 Membership Assessment

As mentioned in Section 2.1, the co-occurrence of the long tail entity with another head entity does not necessarily indicate a strong relationship between them. Therefore, not all constructed communities are related to e_l.

This module produces a *membership score* that reflects the confidence in membership of e_l in a community c, $M(e_l, c)$, for each of the constructed communities. The membership in each community is determined independently from the membership in other communities.

We view this problem as a binary classification task where an entity is labeled "member" or "not member" in the community. LONLIES trains a classifier to compute the probability of an entity's membership in the community. To obtain training data, LONLIES retrieves documents from the text corpus where the community entities appear and extracts classification features from the text. Each entity in the community acts as a training data point. We run the trained classification model against the features extracted for e_l. LONLIES runs two classification tools, Stanford Classifier [5] and DeepDive [7], and assigns the higher score to the community.

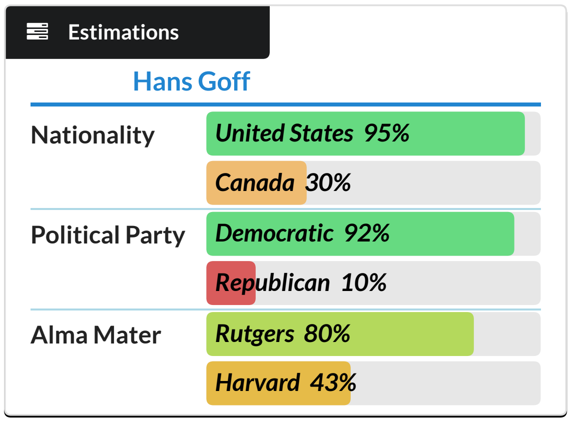
2.4 Estimation

The *Estimation* module receives a set of estimates of p, together with their confidence and membership scores of the communities that produced them. LONLIES uses a straightforward method to aggregate these estimates, using the confidence and membership scores as weights. The result is a distribution of values for p with a final confidence score assigned for each estimate.

3. DEMONSTRATION OVERVIEW

The demonstration audience will be able to interact with the system by issuing queries about properties of long tail entities. The system will show the details of each step in the estimation to explain how it produced the final results. Users can provide feedback about the quality of each step and notice how the system employs the feedback in adjusting the estimation results. LONLIES uses Freebase [2] as its knowledge base and retrieves relevant text documents from Google Search and the ClueWeb09 [3] corpus that is indexed

[1]The index reflects the probability that two entities taken at random from the community have the same p value.


Estimations

Hans Goff

Nationality	United States 95%
	Canada 30%
Political Party	Democratic 92%
	Republican 10%
Alma Mater	Rutgers 80%
	Harvard 43%

Figure 4: Example Estimation Results

by Apache Lucene. We provide a walk-through of an example query and explain what the audience may expect to see.

Users enter two inputs: the name of the long tail entity and properties of interest. As an example, assume that we are interested in the *Nationality*, *Political Party*, and *Alma Mater* of the entity *Hans Goff*, who is a member of Hillary Clinton's presidential campaign. Goff is considered a long tail entity since he does not exist in the knowledge base; however, he is frequently mentioned on the web with other head entities and topics.

Relevant documents and anchors to KB. In the first step, LONLIES retrieves a set of documents that are relevant to the query and provides a list of the anchor KB entities that are extracted from these documents. When the user clicks on an anchor entity, LONLIES shows how many times this head entity appears with the query entity and the list of documents where they appear together, as shown in Figure 2.

Constructed communities and their scores. Given the list of anchor entities, LONLIES constructs a set of communities. Users see a visualization of each community, which is represented as a set of (possibly connected) nodes that are retrieved from the knowledge base. The system presents some statistical details about the community including its confidence of estimating the target properties, the membership score that reflects how much the system believes that the query entity belongs to it, and its size (Figure 2).

When the user clicks on a community, LONLIES shows a detailed view of it. The details include the distribution of values within the community, the features used in determining the membership of entities in this community, and a justification of the produced score of membership.

Users can provide feedback about communities by confirming the membership of the query entity to them or by removing irrelevant communities from the estimation process. The communities are sorted by a score that combines the estimation confidence and the membership of the long tail entity. The higher the rank of a community, the more impact it has on the estimation results. Therefore, it is more valuable to present high-ranking communities to the user and get feedback about most influencing communities.

Estimates. Depending on the type of the property being estimated, LONLIES shows an appropriate visualization of the produced estimates. For categorical properties, LONLIES shows the top-k estimates ordered by their confidence scores (Figure 4 shows the top-2 estimates). For numerical properties, LONLIES shows a distribution of the property being estimated with its mean and variance.

4. RELATED WORK

Knowledge base construction methods, such as DeepDive [7] and Knowledge Vault [4], train property-specific extractors (e.g., using distant supervision) to extract specific relationships from text. Knowledge Vault also exploits a knowledge base to compute the prior of the extracted facts and predict links in the knowledge base graph. This prior is used together with extractions from multiple extraction systems on different data sources to validate the correctness of extracted facts. The success of these approaches relies heavily on the redundancy of information in the text in order to extract accurate property values.

Other question answering techniques [1] utilize query logs to find query-relevant web pages and paid crowd-sourcing to manually extract text snippets and author answers from the retrieved web pages.

5. CONCLUSION

We presented LONLIES, a system that estimates property values for long tail entities. LONLIES leverages the head entities that are co-mentioned with a long tail entity in a text corpus, together with information from a knowledge base, in order to construct a set of communities of entities. LONLIES aggregates the estimates that are produced from communities that (1) are relevant to the long tail entity, and (2) have high estimation power of its members, producing a distribution of the target property value.

6. ACKNOWLEDGMENTS

This work was generously supported by Google Faculty Research Award granted to Ihab F. Ilyas in 2014.

7. REFERENCES

[1] M. S. Bernstein, J. Teevan, S. Dumais, D. Liebling, and E. Horvitz. Direct Answers for Search Queries in the Long Tail. In *SIGCHI*, 2012.

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

[3] J. Callan, M. Hoy, C. Yoo, and L. Zhao. Clueweb09 data set, 2009.

[4] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge Vault: A Web-scale Approach to Probabilistic Knowledge Fusion. In *SIGKDD*, 2014.

[5] C. Manning and D. Klein. Optimization, maxent models, and conditional estimation without magic. In *NAACL - Tutorials '03*, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[6] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2), 2004.

[7] F. Niu, C. Zhang, C. Ré, and J. W. Shavlik. Deepdive: Web-scale knowledge-base construction using statistical learning and inference. *VLDS*, 12:25–28, 2012.

[8] E. H. Simpson. Measurement of diversity. *Nature*, 1949.