A Joint Model for Discovering and Linking Entities

Michael Wick Sameer Singh Harshal Pandya Andrew McCallum
School of Computer Science
University of Massachusetts Amherst MA
{mwick, sameer, harshal, mccallum}@cs.umass.edu

ABSTRACT

Entity resolution, the task of automatically determining which mentions refer to the same real-world entity, is a crucial aspect of knowledge base construction and management. However, performing entity resolution at large scales is challenging because (1) the inference algorithms must cope with unavoidable system scalability issues and (2) the search space grows exponentially in the number of mentions. Current conventional wisdom declares that performing coreference at these scales requires decomposing the problem by first solving the simpler task of entity-linking (matching a set of mentions to a known set of KB entities), and then performing entity discovery as a post-processing step (to identify new entities not present in the KB). However, we argue that this traditional approach is harmful to both entity-linking and overall coreference accuracy. Therefore, we embrace the challenge of jointly modeling entity-linking and entity-discovery as a single entity resolution problem. In order to achieve scalability we (1) present a model that reasons over compact hierarchical entity representations, and (2) propose a novel distributed inference architecture that does not suffer from the synchronicity bottleneck which is inherent in map-reduce architectures. We demonstrate that more test-time data actually improves the accuracy of coreference, and show that joint coreference is substantially more accurate than traditional entity-linking, reducing error by 75%.

1. INTRODUCTION

Wikipedia is a valuable resource because it provides useful information about millions of the world's prominent entities. Recent projects such as Freebase, DBPedia, and Yago have begun enriching Wikipedia's content with formal relational structures (e.g., ontologies and taxonomies of entity types and relationships). As a result, these databases (and the records in them) have become standard touchstones for identifying entities and relations mentioned across the web (e.g., in blogs, newswire articles, personal homepages). For example, newswire articles and blogs frequently discuss entities for which a Wikipedia entry exists (e.g., "Barack Obama"), and will sometimes provide links from the raw textual mentions of these entities to their corresponding Wikipedia or Freebase page. This is a beneficial trend because having links from these mentions to entities opens the possibility of complex semantic queries and pattern analysis over the world's data.

However, the ability to provide comprehensive support for such analysis is currently limited because (1) most of the web's data does not already provide links to these entity records, and (2) Wikipedia and its structured derivatives only contain a small fraction of the world's entities (thus limiting their applicability as a central hub for the world's data). The first problem is addressable via *entity linking*, the task of aligning entities from a database (or noun-phrases from a corpus of newswire text) to a known set of target entities. However, the second problem requires *entity discovery*, which is a more difficult task because the entities are not known *a priori* and must be discovered automatically.

Unfortunately performing these tasks at web-scale is difficult because (1) not all the mentions fit in memory at once, (2) map-reduce architectures are not suitable for entity resolution algorithms and (3) the size of the search space grows exponentially with the number of mentions. As a result, current approaches focus primarily on the easier task of entity-linking, depend heavily on greedy streaming algorithms for inference, and perform entity-discovery (or "nil clustering") only as a post processing step after linking. However, we contend that we can significantly improve the accuracy of both entity-linking and discovery by solving them jointly and by using more data (i.e., gathering more mentions).

In this paper we address the problems of *entity discovery* and *entity linking* jointly. We achieve scalability through two recent innovations. First, we adopt a rich hierarchical representation of entities that compresses their mentions into trees [9, 11]. Second, we propose a novel *asynchronous* parallel Markov chain Monte Carlo (MCMC) procedure that is capable of performing efficient statistical inference over this hierarchical entity representation. Experimentally, we evaluate the hypothesis that solving entity-linking and entity-discovery jointly is more accurate than solving entity-linking and entity discovery in isolation. We further find that coreference resolution is more accurate at larger scales than at smaller scales. The implication of this result is that streaming and greedy coreference algorithms—which cannot reconsider previous coreference decisions—may harm the long-term accuracy of a knowledge base. Finally, we demonstrate that our system is capable of accurately discovering entities that are not already part of the knowledge base.

AKBC CIKM 2013 Workshop on Automated Knowledge Base Construction

2. PROBLEM DESCRIPTION

In this section we describe the problems of entity-linking, entity-discovery and present a more general formulation of entity resolution (coreference) which subsumes the problems of entity-linking and entity-discovery.

2.1 Entity Linking

Entity linking is the problem of matching an entity with all of its referent mentions. More specifically, given a set of known entities $\mathcal{K}$ and a set of mentions $\mathcal{M}$, the problem of entity linking is to output a many-to-one matching from $\mathcal{M}$ to $\mathcal{K}$ such that each mention $m \in \mathcal{M}$ is matched to its corresponding entity $e \in \mathcal{K}$ (if it exists in $\mathcal{K}$) or matched to "nil" otherwise. For example, the set of known entities $\mathcal{K}$ might the set of people/organization/location entities in Wikipedia, and the set of mentions $\mathcal{M}$ might be the set of proper nouns extracted from a collection of newswire articles or blogs. The goal would then be to match the extracted proper nouns in the newswire articles with the Wikipedia entities to which they refer. In this case we would hope to match a mention with the surface form "President Obama" to Obama's Wikipedia page.

2.2 Entity Discovery

Entity discovery (or "nil clustering") is the task of clustering mentions into sets such that all the mentions in a given set all refer to the same real-world entity. The task is similar to entity-linking, except it is more difficult because there are no known entities ($\mathcal{K}$). Entity discovery is often a necessary post-processing step to entity-linking because typically many of the mentions in $\mathcal{M}$ do not have a corresponding entity in $\mathcal{K}$. It is therefore desirable to discover these missing entities by appropriately clustering their mentions.

2.3 Joint Entity Resolution (Coreference)

Entity resolution is an umbrella term encompassing both entity-linking and entity discovery. In this paper, we pose the tasks of entity-linking and entity-discovery as a joint coreference problem. Rather than assuming a set of pre-known entities $\mathcal{K}$, we instead assume only a set of mentions $\mathcal{M}$. Any pre-known entities (for example, Wikipedia pages) are simply treated as mentions (albeit with particularly comprehensive context and high-quality canonical names) and included in the set $\mathcal{M}$. Since we do not assume that we observe the true entities, we instead represent the entities as latent variables in a probabilistic model and infer them with statistical inference[1]. We describe our model and inference procedure in the next section.

3. HIERARCHICAL ENTITY RESOLUTION

In hierarchical entity resolution, the model recursively structures the inferred entities into trees. The leaves of each tree are the entity's mentions, and the non-leaf nodes in the tree recursively summarize the attributes of their children. Thus, the root of each tree is a canonical representation of the entity's attributes which has been inferred from all the entity's mentions. In contrast to traditional pairwise models that measure coreference compatibility between mention pairs, the hierarchical entity resolution model measures coreference compatibility between a child and its parent. Furthermore, since entities and their attributes are random variables in this model, we can also include compatibility functions that measure the cohesiveness of an entity's attributes. We illustrate the hierarchical model instantiated on two entities in Figure 2.

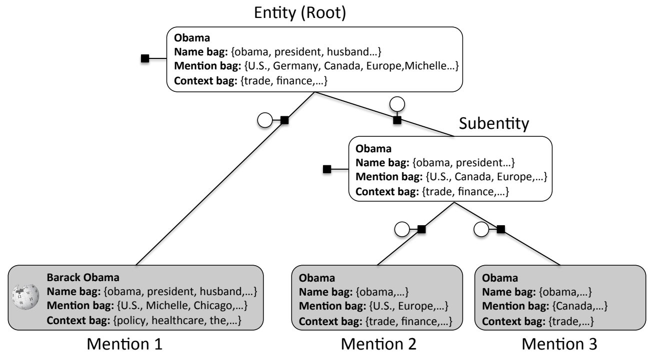

Figure 1: **Hierarchical coreference model** (depicted as a factor graph) instantiated over three mentions with one inferred subentity, and one inferred entity. Black squares represent the compatibility function, shaded boxes represent mentions, and white boxes represent inferred entities/subentities.

In order to perform coreference with the hierarchical model, we use temperature-regulated Markov chain Monte Carlo (MCMC) to search for set of trees that jointly maximizes all the compatibility functions. MCMC explores the search space by iteratively making local improvements to a current coreference hypothesis. For example, MCMC might move a subtree from one entity to another, or propose to create a new entity, or propose to delete a node from a tree. These proposals are then accepted or rejected as a function of how much the model score increased or decreased (due to the proposal). To encourage efficient samples, we make use of pre-defined, high-recall partitioning over entities (called *canopies*) when selecting entities to compare to each other (this is similar to *blocking* used in related work). For more details, see Wick et al. [11].

4. DISTRIBUTED INFERENCE

One of the reasons such joint models are often not used in practice is that the inference problem is considerably difficult to scale. In particular, since each MCMC sample depends on the previous sample, MCMC is an inherently sequential algorithm, and is non-trivial to distribute Recent work by Singh et al. [9] has proposed a Map-Reduce based distributed sampling algorithm that exploits the Markov neighborhood properties of the entity resolution model to scale to millions of mentions. However, the iterated Map-reduce framework faces significant synchronization bottlenecks due to difficulty of load balancing, which is exacerbated for large-scale entity resolution since the size of the entities (and therefore time to compute each sample for them) varies significantly across the dataset (in particular, it often follows the power law, as shown in Singh et al. [10]).

We extend this work to perform distributed inference for the hierarchical entity resolution model in an asynchronous manner. The framework for scaling inference for a large number of mentions consists of the entity features stored

[1]Modeling entities in this way is potentially useful because in practice the known entities (e.g., Wikipedia pages) may be missing some of the attribute values which can be inferred from the other mentions in the dataset.

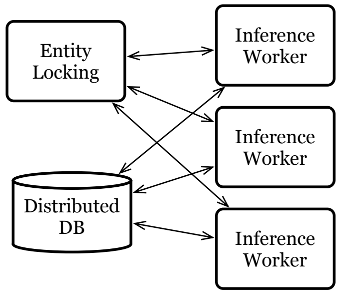

Figure 2: **Asynchronous Distributed Inference:** Entity Locking is a lightweight index that maintains the locking status of each entity. Each inference worker requests locks and reads/writes to the DB completely asynchronously.

in a distributed persistence layer (such as Mongo), and a light-weight entity locking mechanism (essentially an index over entity Ids, often fits in memory on a single machine). Each inference worker asynchronously requests the locking mechanism for a set of entity Ids that are available for inference, and performs inference on them (reading/writing the mention data from/to the distributed DB). This locking mechanism can prioritize different entities for more efficient sampling, for example it is *canopy*-aware in that it assigns entities from within a canopy to a worker. Since each inference worker requests a mutually exclusive set of Ids (ensured by the locking mechanism), there is no contention at the DB level, and the database can efficiently read and write the entities simultaneously. The main bottleneck in this framework is the synchronized entity locking mechanism, however the time spent in requesting locks is much shorter than the time to read/write to the database and the time to perform inference. Nonetheless, if required, a disk-based locking mechanism (such as Redis) may be used, or, for massive-scale resolution, a distributed Hash may also be employed. Using this asynchronous distribution scheme, we are able to scale joint entity discovery and linking to millions of mentions.

5. EXPERIMENTS

5.1 Data

For our experiments, we use the Wikilinks dataset[10] in combination with Wikipedia. Wikilinks is a collection of blogs that contain hyper-links to Wikipedia pages. The anchor texts of these hyper-links are treated as mentions, and the Wikipedia page to which they link is treated as the "ground-truth" entity to which the mention refers. For each Wikilinks mention we create a record of the context that contains various attributes including (1) a bag-of-(context)-words of the tokens in the blog from which it was extracted (2) a bag-of-(mention)-words of the tokens from other mentions in the blog (as identified by a named entity recognition tool), and (3) a bag-of-(name)-words containing the tokens that appear in the surface form of the mention's anchor text.

We also process Wikipedia in a similar fashion. First, we employ the Freebase type hierarchy to identify the person, organization, and location entities in Wikipedia. We extract each of these Wikipedia pages as a *mention* of a real-world entity, which we populate with a set of features that are homologous to those that we extract for Wikilink mentions. In particular, each Wikipedia mention contains (1) a bag-of-(context)-words of the tokens from the Wikipedia page (2) a bag-of-(mention)-words of other anchor texts that appear in that page, and (3) a bag-of-(name)-words consisting of all the tokens in the Wikipedia title plus all the tokens from anchor texts of other Wikipedia pages that link to this page. For example, if Michelle Obama's Wikipedia page were to link to Barack Obama's Wikipedia page via the anchor text "husband," then we would extract "husband" as additional context for the Barack Obama Wikipedia mention.

For the purpose of our experiments, we identity two particularly ambiguous subsets of the combined Wikilinks and Wikipedia data. Specifically, we create one dataset of consisting entirely of "Boston" related organizations and another dataset consisting entirely of "New York" related organizations. The Boston dataset contains all the Wikilinks and Wikipedia mentions that refer to the following Wikipedia entities: Boston (the city itself), the Boston Celtics (professional basketball team), the Boston Red Sox (professional baseball team), the Boston Bruins (professional hockey team), and the Boston Globe (newspaper). The New York dataset includes: the New York Yankees (baseball), the New York Knicks (basketball), the New York Rangers (hockey), the New York Giants (Football), and the New York Jets (also Football). Each dataset has approximately 5000 mentions, and each entity has between 500 and 1800 mentions.

We chose these two subsets because they are especially challenging: organizations that are named after the cities to which they belong are ambiguous since they have similar context and overlapping names (e.g., the names of the organizations contain the words "Boston" and "New York" respectively). Furthermore, it is common practice in blogs to refer to a particular sports organization simply by the name of the city from which they are based. For example, "Boston" could refer to the "Boston Celtics," the "Boston Red Sox," or the "Boston Bruins" depending on the context. Additionally, sports teams often have overlapping context words such as "beat," "goal," and "score." Finally, sports organizations tend to have many nicknames. For example, the "New York Yankees" are also known as the "Bronx Bombers" and the "NY Highlanders," and "Boston" is also known as "Beantown." In comparison, people and most other organizations are on average significantly easier.

5.2 Systems and baselines

As in previous work by Wick et al. [11], we manually set the parameters. For these experiments we tune the parameters on the Boston dataset, and use the New York dataset to evaluate the coreference systems and baselines. In particular, we evaluate the following systems:

String-match: this system clusters all mentions that have the same canonical name string.

Entity-linking (streaming-k) same as above, except instead of using MCMC for inference, it makes k passes over all the Wikilinks mentions. It visits each mention (one at a time) and attempts to merge it with the Wikipedia entity for which it has the highest model score (or none if all the scores are negative). A value of $k = 1$ is the traditional streaming setting where the system must make one decision for each mention before moving on to the next [7]. A higher value of k allows the system to revisit an old decision which could be more accurate since more mention context has been aggregated in the entity.

Entity-linking (MCMC) the entity-linking system treats

Method	PW F1	Link Acc.
String matching baseline	83.6	91.3
Entity linking (streaming x1)	83.7	92.0
Entity linking (streaming x2)	83.9	92.2
Entity linking (streaming x4)	84.0	92.2
Entity linking (MCMC)	84.0	92.2
Joint linking+discovery	**97.3**	**98.2**

Table 1: **Evaluation of Linking and Discovery**

Pre-known Entities witheld	PW F1
None	97.342
only NY Yankees	96.6
only NY Rangers	96.7
only NY Knicks	96.9
only NY Giants	89.5
only NY Jets	89.1
All	89.776

Table 2: Evaluating the ability to discover entities, when the various pre-known (Wikipedia) entities are witheld

the Wikipedia mentions as a set of known entities. During inference, the entity-linking systems only considers MCMC moves that would either add or remove a link between a Wikilinks mention and an entity that contains a Wikipedia mention. This system cannot create new entities.

Joint entity-linking+discovery models entity linking and entity discovery jointly and solves the full coreference problem using MCMC (which in contrast to the entity-linking MCMC algorithm, can also consider merging entity trees which do not contain any Wikipedia mentions).

5.3 Results

In this section we evaluate the joint entity-linking and entity discovery approach. First, in Table 1 we compare the joint approach to several commonly employed baselines. We find that solving the full joint coreference problem (evaluating both coreference and entity-linking accuracy) achieves a 75% reduction in error versus the closest approach. This result indicates that current procedures to entity-linking (for example, in TAC-KBP) could be greatly improved by jointly solving the nil-clustering problem rather than deferring it as a post-processing step.

Next, we evaluate our system's ability to perform entity-discovery (that is, coreference of mentions for which we lack known Wikipedia page). We simulate missing entities by withhold Wikipedia pages from the NY dataset and then evaluating our system on the modified data. We report the results in Table 2. Note that some entities are more difficult to discover than others; for example, the system performs worse when withholding one of the two football team's (Jets and Giants) Wikipedia page because the mentions are more contextually similar. However, overall, our system still achieves relatively high accuracy (approximately 90% F1) even when all the Wikipedia pages are withheld.

#Mentions			F1
additional	seed	total	(on seeds)
0	2275	2275	88.4
759	2275	3034	89.8
1518	2275	3793	95.5
2275	2275	4550	96.6

Table 3: Evaluating the effect of additional mentions on the performance of coreference resolution (NY dataset).

Finally, we examine how the number of mentions impacts the accuracy of our coreference resolution system. Note that as the number of mentions increases, the size of the search space grows exponentially making coreference more difficult. However, the amount of available information about each entity also increases which should on the other hand have the effect of making coreference easier. In this experiment, we evaluate coreference accuracy on a fixed subset of the mentions, but vary the number of additional mentions input to coreference. Table 3 shows that adding additional mentions helps coreference more accurately resolve the fixed set of seed mentions. This result highlights the importance of building scalable coreference systems.

6. RELATED WORK

There are a number of different approaches to large-scale coreference resolution. Entity-linking systems, which include Wikifiers (systems that resolve mentions against Wikipedia) [5, 8], solve a simpler formulation of coreference in which the entities are already known (i.e., provided by a knowledge base such as Wikipedia) and the task is to link mentions to this fixed set of provided entities. Record-linking systems [3, 6, 2], which disambiguate records of entities *across* databases (but not *within* each database), relax the assumption of a fixed set of entities; however, they usually assume that each database has already been disambiguated [2]. Thus, entity-linking and record-linking have limited utility because the former cannot discover the existence of new entities and the latter can only incorporate entities from databases which have previously been disambiguated. In contrast, we address a more widely applicable formulation of the coreference problem in which (1) entities are not assumed to be known in advance and (2) each dataset is not assumed to be disambiguated.

There has also been work in addressing the full cross-document coreference problem. These approaches, including ours, typically employ some form of blocking [1] or canopies [4], techniques for reducing the search space by partitioning the mentions into overlapping sets such that mentions that never appear in the same set need not be considered for coreference. However, blocking alone is not sufficient for scalability and there has been a variety of proposed techniques for addressing this issue including formulating coreference as a streaming inference problem [7], reducing the number of similarity functions via single-link agglomerative clustering [2], and compressing the data by averaging the feature vectors of mentions which refer to the same entities [2, 7]. Although streaming approaches are highly scalable, they suffer from permanently low accuracy because all coreference decisions are final (they are not able to use the information provided in later mentions to retroactively correct coreference errors for old mentions). The problem with approaches that compress the data by averaging feature vectors is that they sacrifice representational power crucial for resolving highly ambiguous mentions.

7. CONCLUSIONS

In this paper we presented a scalable solution for solving entity-linking and entity-discovery jointly. First, we demonstrated that solving the full joint coreference resolution problem results in higher accuracy than just solving entity-linking in isolation. We also showed that including more mentions actually improves coreference accuracy. Finally, we evaluated our system on the problem of entity-discovery and demonstrated that it predicts new entities with high accuracy.

References

[1] M. Bilenko, B. Kamath, and R. J. Mooney. Adaptive blocking: Learning to scale up record linkage. In *Proceedings of the Sixth International Conference on Data Mining*, ICDM '06, pages 87–96, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2701-9. doi: http://dx.doi.org/10.1109/ICDM.2006.13. URL `http://dx.doi.org/10.1109/ICDM.2006.13`.

[2] C. Bohm, G. de Melo, F. Naumann, and G. Weikum. Record linking: the design of efficient systems for linking records into individual and family histories. In *CIKM*, 2012.

[3] H. L. Dunn. Record linkage. *American Journal of Public Health*, 36(12):1412–1416, 1946.

[4] A. K. McCallum, K. Nigam, and L. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the Sixth International Conference On Knowledge Discovery and Data Mining (KDD-2000)*, Boston, MA, 2000.

[5] R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, CIKM '07, pages 233–242, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-803-9.

[6] H. B. Newcombe. Linda: Distributed web-of-data-scale entity matching. *the American Journal of Human Genetics*, 19(3):334–359, 1967.

[7] D. Rao, P. McNamee, and M. Dredze. Streaming cross document entity coreference resolution. In *COLING (Posters)*, pages 1050–1058, 2010.

[8] L. Ratinov, D. Roth, D. Downey, and M. Anderson. Local and global algorithms for disambiguation to wikipedia. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2011. URL `http://cogcomp.cs.illinois.edu/papers/RRDA11.pdf`.

[9] S. Singh, A. Subramanya, F. Pereira, and A. McCallum. Large-scale cross-document coreference using distributed inference and hierarchical models. In *Association for Computational Linguistics: Human Language Technologies (ACL HLT)*, 2011.

[10] S. Singh, A. Subramanya, F. Pereira, and A. McCallum. WikiLinks: Large-scale cross-document coreference corpus labeled via links to wikipedia. Technical Report UM-CS-2012-015, University of Massachusetts, Amherst, 2012.

[11] M. Wick, S. Singh, and A. McCallum. A discriminative hierarchical model for fast coreference at large scale. In *Association for Computational Linguistics (ACL)*, 2012.