A Context-Aware Approach to Entity Linking

Veselin Stoyanov and **James Mayfield**
HLTCOE
Johns Hopkins University

Tan Xu and **Douglas W. Oard**
University of Maryland
College Park

Dawn Lawrie
Loyola University
in Maryland

Tim Oates and **Tim Finin**
University of Maryland
Baltimore County

Abstract

Entity linking refers to the task of assigning mentions in documents to their corresponding knowledge base entities. Entity linking is a central step in knowledge base population. Current entity linking systems do not explicitly model the discourse context in which the communication occurs. Nevertheless, the notion of shared context is central to the linguistic theory of pragmatics and plays a crucial role in Grice's cooperative communication principle. Furthermore, modeling context facilitates joint resolution of entities, an important problem in entity linking yet to be addressed satisfactorily. This paper describes an approach to context-aware entity linking.

1 Introduction

Given a mention of an entity in a document and a set of known entities in a knowledge base (KB), the *entity linking* task is to find the entity ID of the mentioned entity, or return NIL if the mentioned entity was previously unknown. Entity linking is a key requirement for knowledge base population; without it, accurately extracted attributes and relationships cannot be correctly inserted into an existing KB.

Recent research in entity linking has been driven by shared tasks at a variety of international conferences (Huang et al., 2008; McNamee and Dang, 2009). The TAC Knowledge Base Population track (Ji et al., 2011) provides a representative example. Participants are provided with a knowledge base derived from Wikipedia Infoboxes. Each query comprises a text document and a mention string found in that document. The entity linking system must determine whether the entity referred to by the mention is represented in the KB, and if so, which entity it represents.

State-of-the-art entity linking systems are quite good at linking person names (Ji et al., 2011). They rely on a variety of Machine Learning approaches and may incorporate different external resources such as name Gazetteers (Burman et al., 2011), precompiled estimates of entity popularities (Han and Sun, 2011) and modules trained to recognize name and acronym matches (Zhang et al., 2011).

Two areas are handled less well by current entity linking systems. First, it has been recognized that collective inference over a set of entities can lead to better performance (Cucerzan, 2007; Kulkarni et al., 2009; Hoffart et al., 2011; Ratinov et al., 2011). While the field has begun to move in the direction of collective (or joint) inference, such inference is a computationally hard problem. As a result, current joint inference approaches rely on different heuristics to limit the search space. Thus, collective classification approaches are yet to gain wide acceptance. In fact, only four of the 35 systems that submitted runs to the 2011 TAC KBP task go beyond a single query in a single document. Ji et al. (2011) cite the need for (more) joint inference as one of the avenues for improvement.

The second area not handled well is the notion of discourse context. Grice's principle for collaborative communication postulates that communications should obey certain properties with respect to the context shared between the author and the recipient of the communication (Grice, 1975). For instance, the Maxim of Quantity states that a contribution should be as informative as is required (for the purpose of the exchange), but no more informative than that. Similarly, the Maxim of Manner states that one should avoid ambiguity. Grice's principle is important for entity linking: it argues that communications (e.g., newswire articles) are only possible when the author and the audience share a discourse context, and entity mentions must be unambiguous in this shared context.

The shared discourse context depends on the type of communication, the author, and the intended audience. For newswire with a given readership, there is a broadly shared context, comprising the major personalities and organizations in politics, sports, entertainment, etc. Any entity mentioned that is not part of this broadly shared context will be fully qualified in a news article (e.g., *"Jane Frotzenberry, 42, a plumber from Boaz, Alabama said ... "*). Thus, a system that performs entity linking on newswire needs to maintain a list of entities that are famous at the given time. Less famous entries can be resolved with the help of the extra information that the author provides, as required by the Maxim of Quantity.

The notion of context is all the more important when resolving entities in personal communications such as email. Personal communication often contains unqualified entity mentions. For example, an email from **Ken Lay** to **Jeff Skilling** might mention *Andy* with no other indication that the person mentioned is **Andrew Fastow**. A traditional entity linking system will fail miserably here; the mention *Andy* is simply too ambiguous out of context. Email-specific linkers often rely on access to the communications graph to resolve such mentions. The communications graph is important mainly because it offers a guess at the discourse context shared between the author of a communication and its recipient(s).

We propose a new approach to entity linking that explicitly models the context shared by the participants of a communication. Our context-aware entity linking approach is guided by three principles:

1. *Shared context should be modeled explicitly.* This allows the linker to be easily adapted to new genres, and allows a modular system design that separates context modeling from entity linking.

2. *Most entity linking should be trivial in the shared context.* If the context accurately models the shared assumptions of author and audience, mentions should identify known entities in the context with little ambiguity.

3. *Context facilitates joint inference.* A joint resolution of all entities in a communication must be consistent with a given context. Thus, a resolver must find a context that explains why the particular set of entities are mentioned together. In other words, the discourse context is an extension of the joint resolution of the document's mentions together with additional related entities that are not mentioned in the particular document. Joint context has been recognized as an important notion for collective assignment of a set of mentions (Kulkarni et al., 2009; Ratinov et al., 2011; Hoffart et al., 2011), but previous work has not explicitly modeled the discourse context between the author and recepients of a communication. From a computational point of view the notion of context has two advantages: it limits the number of possibilities that a resolver must consider; and it motivates an efficient iterative joint resolution procedure.

In this paper, we outline a new architecture for context-aware entity linking and discuss our particular implementation. Our system is suitable for both newswire articles and first person communication. We also present some preliminary results.

2 What is a context?

According to linguistic theory, discourse context encompasses the knowledge and beliefs that are shared between the author and the recipient of a communication (Bunt and Black, 2000). This can include objects introduced earlier in the discourse as well as general knowledge that a communication's author can assume the audience possesses.

Representing all knowledge shared between an author and a recipient of a communication is challenging – it requires solving difficult knowledge acquisition and representation problems. We use a more limited notion of context; we define a context to be a weighted set of KB entities. For example, a general *US newswire* context may contain, with high weight, public entities such as **Barack Obama**, **Mitt Romney** and **LeBron James**.

The set of entities that make up a context and their weights should be determined by a number

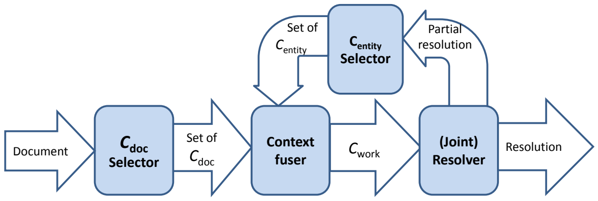

Figure 1: Architecture of our context aware entity linking system.

of factors: the intended audience for a communication (e.g., a typical Westerner vs. a German college student vs. an AKBC-WEKEX 2012 atendee); the time and place of the communication (some entities are only popular over a limited time span); and the topic of the communication (e.g., *Jordan* likely refers to the country when talking about the Middle East; it likely refers to **Michael Jordan** when discussing basketball). Furthermore, the makeup of the context may change as the recipient of communication is provided with more information. For instance, learning that a document talks about **Barack Obama** gives associated entities such as **Joe Biden** and **Michelle Obama** a higher weight.

To accommodate a diverse range of approaches to context, we define a general context-aware architecture that makes few additional assumptions on what contexts can be or how entities can be brought into or re-weighted in the current context. In the next section we describe the general architecture of our system. We then discuss how we generate contexts for newswire and email in Section 4.

3 Context-sensitive Entity Linking Architecture

First we introduce the following terminology to refer to different kinds of context:

- $\mathbf{C}_{work}$ (working context) – the weighted set of entities against which the system is currently resolving mentions. For example, the system may begin with a general context of all prominent entities discussed in the world news. As the system makes decisions about how entities are linked, it may revise the set of entities that are under consideration. The working context can be updated as processing proceeds.

- $\mathbf{C}_{doc}$ (document context) – a context triggered by a particular communication (document). For instance, an article in the New York Times may evoke a particular set of weighted entities. Document contexts can be quite specific; there can be a different document context for each section of the New York Times, for each author, or for each topic of discussion.

- $\mathbf{C}_{entity}$ (entity context) – an entity context refers to the weighted set of entities associated with a particular KB entity. If the system resolves a mention to an entity with high confidence, it updates its working context to include or up-weight these associated entities.

We use *trigger* to refer to a function that given a document or an entity and produces all of the C_{doc}s and C_{entity}s associated with the document or entity respectively. This could be a simple function that keeps an inverted index that associates words with database entities and, for given document, retrieves the entities most associated with the words of the document. It could also be a more sophisticated function that identifies contexts as graph communities and/or observes which entities are often mention together in a corpus of similar communications (e.g., newswire articles). The latter trigger would need either a large corpus of annotated communication or a bootstrapping method to associate entities with communications. Triggers can also associate general contexts with a given source or audience (e.g., a general context associated with the New York Times) or specific contexts associated with the topic of the document (e.g., the IR-based trigger discussed above that associates specific words with each entity and produces a weighted list of matches given the document).

The overall architecture of our context-aware entity linking system is shown in Figure 1. The system processes documents (communications) marked with the mentions that need to be resolved. Processing of a document begins by invoking a collection of *triggers* to produce a set of C_{doc}s associated with the document. Triggers are functions mapping documents to contexts.

The set of selected C$_{doc}$s is then passed to a context fuser. The fuser unifies individual contexts and produces a single set of entities, which becomes C$_{work}$. Different algorithms could be used to fuse entities coming from different contexts; we currently use a simple summation of the weights.

The working context is then fed to a resolver. The job of the resolver is to decide for each mention whether there is an entity that represents a good match for that mention. The resolver can produce partial matches (e.g., decide not to match some mentions or match other mentions to more than one entity) in early iterations, but is required to produce a full match at the final iteration.

The partial match produced by the resolver is fed to a C$_{entity}$ selector, which selects a set of entities related to each resolved mention. The selector produces a set of C$_{entity}$s, which, together with C$_{work}$, are passed again to the context fuser. This process repeats until either all mentions are resolved to a desired level of confidence or a predefined number of iterations is reached. Upon termination, the algorithm returns an entity match for each mention of the document or *NIL* to indicate that no match exists in the knowledge base.

4 Generating contexts

The success of our approach hinges on the ability to generate and later retrieve effective contexts. Our system currently implements simple context triggers, so most of the triggers discussed in this section are subject of future work. Triggers are tailored to the domain of the communication. We are experimenting with two domains: linking newswire articles to Wikipedia pages and linking names in emails to their corresponding email addresses.

C$_{doc}$ generation. For newswire articles, we currently rely on a single IR-based trigger. This trigger uses Lucene[1] to create an index associating words with Wikipedia entities, based on the content of the Wikipedia page associated with the entity. The trigger then queries the index using the first paragraph in which a given entity is mentioned in the document (e.g., if we want to resolve *Clinton*, our query will be the first paragraph in the document mentioning *Clinton*). Some additional C$_{doc}$ triggers that we plan to implement for this domain include: geographic-, time- and source-specific triggers, and evolutionary triggers that are based on resolutions found in previously processed documents. Note that some of these triggers require a corpus of articles linked to KB entities. We are investigating using bootstrapping and other methods to produce triggers. We also plan to use graph partition algorithms to discover communities in the KB, and use those communities as a source of smoothing (since some entities may be infrequently mentioned).

For email, we currently use three C$_{doc}$ triggers: (D1) an IR-based trigger, that retrieves entities according to the text in emails previously sent or received by the entity; (D2) another IR-based trigger that uses entities in the “from,” “to,” and “cc” fields of emails relevant to the query email; and (D3) author-specific contexts based on the communication graph. In future work, we plan to use bootstrapping and community detection to expand our email C$_{doc}$ triggers.

C$_{entity}$ generation. For each entity in the KB, its C$_{entity}$ aims to capture a set of related entities. To determine the degree of entity relatedness in newswire, we use a measure based on network distance and textual relatedness (we currently link against Wikipedia, so the text is harvested from the article associated with the entity).

For email, each C$_{entity}$ consists of all one-hop neighbors in the communication graph in which only entity pairs that have exchanged at least one message in each direction are linked.

In future work, we plan to implement E-contexts that use large unsupervised corpora and bootstrapping to determine which entities tend to occur together in documents. Here, again, we plan to use a graph partition algorithm to discover communities and use those for smoothing.

5 Evaluation

Data. We evaluate our newswire system on the data created for the last three TAC entity linking track (McNamee and Dang, 2009; Ji et al., 2010; Ji et al., 2011). This data consists of 6,266 query mentions over 5,962 documents. The KB is formed from the infoboxes of a Wikipedia dump. For email, we use the Enron collection (Klimt and Yang, 2004).

[1] http://lucene.apache.org

Ground truth is given by the publicly available set of 470 single-token mentions (in 285 unique emails) that have been manually resolved to email addresses by Elsayed (2009).

Evaluation metrics. We evaluate two components of our system – the working context (C_{work}) and the resolver accuracy. For a working context to be useful for our task, it has to include the gold-standard entities against which mentions in a document are resolved. Thus, we evaluate the working context by its recall, computed as the number of gold-standard entities in the context divided by the total number of entities to be resolved (excluding NILs). Overall system performance is compared on the accuracy of the final resolution of all mentions (including those that are assigned a *NIL* in the gold standard).

Results. Results presented here are preliminary: we currently use simple string-match based resolvers and incorporate only a subset of the contexts that we intend to implement.

On newswire, we rely on a parameter that sets the maximum number of entities returned by the trigger. When we set the parameter to 500, the context recall on non-NIL is 0.735 and the average number of entities per document returned is 452 (some documents return less than the maximum number, 500). When we set the parameter to 5,000, the context recall on non-NIL is 0.829 and the average number of entities is 4,515. We contrast this to the triage mechanism of McNamee et al. (2011), which relies on name and alias matching to obtain all potential entity matches. This mechanism achieves recall of 0.941 on non-NIL with average context size of 6,190. The set of entities returned by the triage mechanism are much most ambiguity as all of the entities in the set share the same name or alias (or character n-grams found in the mention).

The overall accuracy of the system in the two settings that rely on our document trigger is around 0.6 in both settings (including NILs), while the accuracy of the system using McNamee et al.'s (2011) triage is around 0.3 (including NILs). As discussed above, we currently use a simple rule-based string matching resolver. Additionally, most of the TAC queries ask for one mention per document, so on newswire our system cannot take full advantage of the C_{entity} mechanism. We are working on expanding the query set to include additional unsupervised mentions that are resolved but not scored.

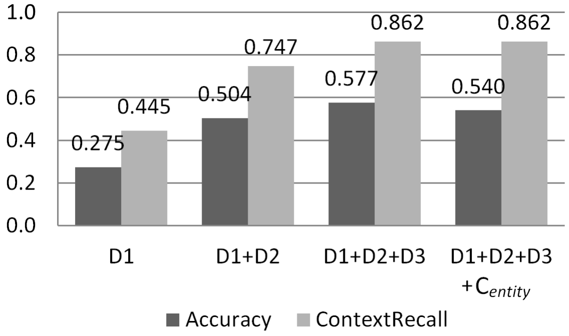

Figure 2: Enron dataset results.

Results for email are shown in Figure 2. We use three different document triggers (described in the previous section). Results show that our simple context fuser effectively leverages multiple C_{doc}s, but a more sophisticated resolver to optimally exploit both C_{doc}s and C_{entity}s is needed.

6 Conclusions

We argue that the notion of discourse context is central to entity linking, and that it facilitates joint inference. We introduce a system that performs context-aware entity linking by building a working context from document and entity contexts. The working context is refined during the course of linking mentions in a communication so that all entities can be linked with high confidence.

References

Harry Bunt and William Black, 2000. *The ABC of computational pragmatics*. MIT Press.

A. Burman, A. Jayapal, S. Kannan, M. Kavilikatta, A. Alhelbawy, L. Derczynski, and R. Gaizauskas. 2011. USFD at KBP 2011: Entity linking, slot filling and temporal bounding. *Proceedings of TAC 2011*.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of EMNLP-CoNLL*, pages 708–716.

Tamer Elsayed. 2009. *Identity resolution in email collections*. Ph.D. thesis, University of Maryland.

Paul Grice. 1975. Logic and conversation. *Syntax and semantics*, 3:41–58.

X. Han and L. Sun. 2011. A generative entity-mention model for linking entities with knowledge base. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 945–954.

J. Hoffart, M.A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of EMNLP*, pages 782–792.

Darren Wei Huang, Yue Xu, Andrew Trotman, and Shlomo Geva. 2008. Overview of INEX 2007 Link the Wiki track. In Norbert Fuhr, Jaap Kamps, Mounia Lalmas, and Andrew Trotman, editors, *Focused Access to XML Documents*, pages 373–387. Springer-Verlag, Berlin, Heidelberg.

H. Ji, R. Grishman, H.T. Dang, K. Griffitt, and J. Ellis. 2010. Overview of the TAC 2010 knowledge base population track. *Proceedings of Text Analysis Conference (TAC)*.

H. Ji, R. Grishman, and H.T. Dang. 2011. Overview of the TAC 2011 knowledge base population track. *Proceedings of Text Analysis Conference (TAC)*.

Bryan Klimt and Yiming Yang. 2004. The Enron corpus: A new dataset for email classification research. In *ECML*, pages 217–226.

S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. 2009. Collective annotation of wikipedia entities in web text. In *Proceedings of ACM SIGKDD*, pages 457–466.

P. McNamee and H.T. Dang. 2009. Overview of the TAC 2009 knowledge base population track. In *Proceedings of Text Analysis Conference (TAC)*.

P. McNamee, J. Mayfield, D.W. Oard, T. Xu, K. Wu, V. Stoyanov, and D. Doermann. 2011. Cross-language entity linking in maryland during a hurricane. In *Proceedings of Text Analysis Conference (TAC)*.

L. Ratinov, D. Downey, M. Anderson, and D. Roth. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of ACL*.

Wei Zhang, Yan Chuan Sim, Jian Su, and Chew Lim. 2011. Entity linking with effective acronym expansion, instance selection and topic modeling. *Proceedings of IJCAI*.