Ontology-Driven Automatic Entity Disambiguation in Unstructured Text.
(To Appear in) Proceedings of ISWC-2006, LNCS

Ontology-Driven Automatic Entity Disambiguation in Unstructured Text

Joseph Hassell, Boanerges Aleman-Meza & I. Budak Arpinar

Large Scale Distributed Information Systems (LSDIS) Lab
Computer Science Department, University of Georgia
Athens, GA 30602-7404, USA
{hassell, boanerg, budak}@cs.uga.edu

Abstract. Precisely identifying entities in web documents is essential for document indexing, web search and data integration. Entity disambiguation is the challenge of determining the correct entity out of various candidate entities. Our novel method utilizes background knowledge in the form of a populated ontology. Additionally, it does not rely on the existence of any structure in a document or the appearance of data items that can provide strong evidence, such as email addresses, for disambiguating person names. Originality of our method is demonstrated in the way it uses different relationships in a document as well as from the ontology to provide clues in determining the correct entity. We demonstrate the applicability of our method by disambiguating names of researchers appearing in a collection of DBWorld posts using a large scale, real-world ontology extracted from the DBLP bibliography website. The precision and recall measurements provide encouraging results.

Keywords: Entity disambiguation, ontology, semantic web, DBLP, DBWorld.

1 Introduction

A significant problem with the World Wide Web today is that there is no explicit semantic information about the data and objects being presented in the web pages. Most of the content encoded in HTML format serves its purpose of describing the presentation of the information to be displayed to human users. HTML lacks the ability to semantically express or indicate that specific pieces of content refer to real-world named entities or concepts. For instance, if “George Bush” is mentioned on a web page, there is no way for a computer to identify which “George Bush” the document is referring to or even if “George Bush” is the name of a person.

The Semantic Web aims at solving this problem by providing an underlying mechanism to add semantic metadata on any content, such as web pages. However, an issue that the Semantic Web currently faces is that there is not enough semantically annotated web content available. The addition of semantic metadata can be in the form of an explicit relationship from each appearance of named entities within a document to some identifier or reference to the entity itself. The architecture of the Semantic Web relies upon URIs [4] for this purpose. Examples of this would be the entity “UGA” pointing to http://www.uga.edu and “George Bush” pointing to a URL of his official web page at the White House. However, more benefit can be obtained by referring to actual entities of an ontology where such entities would be related to concepts and/or other entities. The problem that arises is that of entity disambiguation, which is concerned with determining the right entity within a document out of various possibilities due to same syntactical name match. For example, “A. Joshi” is ambiguous due to various real-world entities (i.e. computer scientists) having the same name.

Entity disambiguation is an important research area within Computer Science. The more information that is gathered and merged, the more important it is for this information to accurately reflect the objects they are referring to. It is a challenge in part due to the difficulty of exploiting, or lack of background knowledge about the entities involved. If a human is asked to determine the correct entities mentioned within a document, s/he would have to rely upon some background knowledge accumulated over time from other documents, experiences, etc. The research problem that we are addressing is how to exploit background knowledge for entity disambiguation, which is quite complicated particularly when the only available information is an initial and last name of a person. In fact, this type of information is already available on the World Wide Web in databases, ontologies or other forms of knowledge bases. Our method utilizes background knowledge stored in the form of an ontology to pinpoint, with high accuracy, the correct object in the ontology that a document refers to. Consider a web page with a “Call for Papers” announcement where various researchers are listed as part of the Program Committee. The name of each of them can be linked to their respective homepage or other known identifiers maintained elsewhere, such as the DBLP bibliography server. Our approach for entity disambiguation is targeted at solving this type of problem, as opposed to entity disambiguation in databases which aims at determining similarity of attributes from different database schemas to be merged and identifying which record instances refer to the same entity (e.g., [7]).

The contributions of our work are two-fold: (1) a novel method to disambiguate entities within unstructured text by using clues in the text and exploiting metadata from an ontology; (2) an implementation of our method that uses a very large, real-world ontology to demonstrate effective entity disambiguation in the domain of Computer Science researchers. According to our knowledge, our method is the first work of its type to exploit an ontology and use relations within this ontology to recognize entities without relying on structure of the document. We show that our method can determine the correct entities mentioned in a document with high accuracy by comparing to a manually created and disambiguated dataset.

2 Dataset

Our dataset consists of two parts. First, an ontology created from the DBLP bibliography [14] and a corpus of DBWorld documents [6] that we use to evaluate our system. We chose the DBLP dataset because it is a rich source of information in the Computer Science domain and DBWorld because it contains documents which include names of people that typically exist in DBLP.

2.1 DBLP

Our goal is to demonstrate real-world applicability of our approach. Therefore, we chose to use data from the DBLP bibliography site (which has been around since the 1980's). This is a web site that contains bibliographic information for computer science researchers, journals and proceedings. Currently, it indexes more than 725,000 articles and contains a few thousand links to home pages of computer scientists. Conveniently, the site provides two XML files that contain most of the information stored in its servers. One of the files contains objects such as authors, proceedings and journals. The other file contains lists of papers usually organized by tracks or sessions of the conference or workshop where they were presented. We have taken the former and converted it into RDF. The resulting RDF is very large, approximately one and a half gigabytes. It contains 3,079,414 entities and 447,121 of these are authors from around the world. Table 1 lists the classes with the most instances.

Table 1. Instances of classes in DBLP ontology

Authors	447,121
Journal Articles	262,562
Articles in Proceedings	445,530

The conversion to RDF was designed to create entities out of peoples' names, instead of treating the names as literal values being part of the metadata of a publication. For this reason, we did not make use of other available RDF-converted data of DBLP (e.g., http://www.semanticweb.org/library/#dblp). Additionally, the data in RDF is enriched by adding relationships to affiliations (i.e., universities) and research topics for researchers. For further details see http://lsdis.cs.uga.edu/projects/semdis/swetodblp/.

2.2 DBWorld

DBWorld is a mailing list of information for upcoming conferences related to the databases field. Although it does contain some random post about open positions, etc., we are only interested in postings about conferences, workshops, and symposiums.

We created an HTML scraper that visits the DBWorld site and downloads only the posts that contain "Call for Papers", "Call for Participation" or "CFP" in the subject. Our system disambiguates the people listed in these postings and provides a URI to the corresponding entity in the ontology.

A DBWorld post typically contains an introduction, topics of interest, important dates and a list of committee members. The general layout of the DBWorld post is rarely consistent in terms of its structure. For example, sometimes the participants of a conference are listed with their school or company affiliation and sometimes they are listed along with the name of a country.

3 Approach

In our approach, different relationships in the ontology provide clues for determining the correct entity out of various possible matches. Figure 1 provides an overview of the main modules in our approach. We argue that rich semantic metadata representations allow a variety of ways to describe a resource. We characterize several relationship types that we identified and explain how they contribute towards the disambiguation process. As mentioned, we use the scenario of disambiguating researchers by their names appearing in DBWorld postings. However, we believe that the following relationship types are applicable to other scenarios (such as disambiguating actor names in movie reviews).

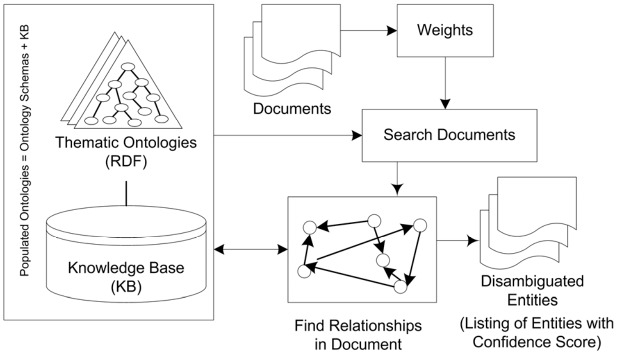


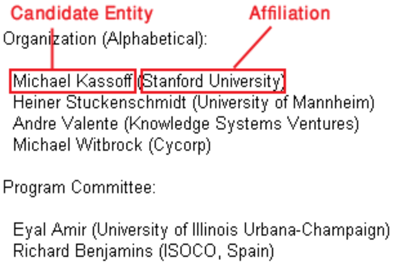
Fig. 1. Overview of the Main Modules for Entity Disambiguation

3.1. Entity Names

An ontology contains a variety of concepts and instance data. The first step of our approach is specifying which entities from a populated ontology are to be spotted in text and later disambiguated. To do this, it is necessary to indicate which literal property is the one that contains the 'name' of entities to be spotted. In most cases, such a literal property would be 'rdfs:label.' However, in some cases, additional 'name' properties may need to be listed, such as aliases and alternate names. Additionally, a different ontology may have its own way of representing the name for each entity.

3.2 Text-proximity Relationships

Various relationships contain metadata that can be expected to be in 'text-proximity' of the entity to be disambiguated. For example, affiliation data commonly appears near names of researchers in DBWorld posts. Hence, when the known affiliation (from the ontology) appears near an entity, there is an increased likelihood that this entity is the correct entity that the text refers to. This 'nearness' is measured by the number of space characters between two objects. Figure 2 illustrates an example where the affiliation "Stanford University" appears next to the entity of interest, "Michael Kassoff", whose known affiliation is "Stanford University" according to the populated DBLP ontology. We acknowledge the fact that the *up to date* status of an ontology can have an impact on the quality of disambiguation results yet measuring the degree of such impact is outside the scope of this paper.



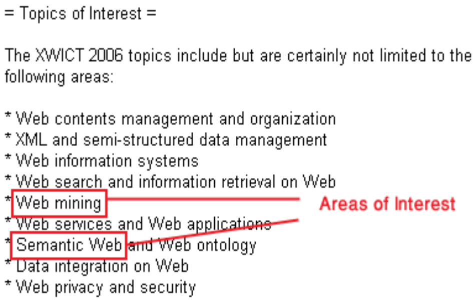
Organization (Alphabetical):

Michael Kassoff (Stanford University)
Heiner Stuckenschmidt (University of Mannheim)
Andre Valente (Knowledge Systems Ventures)
Michael Witbrock (Cycorp)

Program Committee:

Eyal Amir (University of Illinois Urbana-Champaign)
Richard Benjamins (ISOCO, Spain)

Fig. 2. Snippet from a DBWorld post



= Topics of Interest =

The XWICT 2006 topics include but are certainly not limited to the following areas:

* Web contents management and organization
* XML and semi-structured data management
* Web information systems
* Web search and information retrieval on Web
* Web mining
* Web services and Web applications
* Semantic Web and Web ontology
* Data integration on Web
* Web privacy and security

Fig. 3. Snippet from the same post in Figure 2

3.3 Text Co-occurrence Relationships

Text co-occurrence relationships are similar to text-proximity relationships with the exception that 'proximity' is not relevant. For example, the intuition of using affiliation data is applicable as long as it appears 'near' a person entity, but it would not be relevant if it appears somewhere else in the text because it could be the affiliation of a different person (or referring to something else). Text co-occurrence relationships are intended to specify data items that, when appearing in the same document, provide clues about the correct entity being referred in the text. For example, in DBWorld posts, the listed 'topics' fit the idea of text co-occurrence relationships. Figure 3 shows a portion of the same document in Figure 2, where "Web mining" and "Semantic Web" are spotted and are both areas of interest that match research topics related to "Michael Kassoff." Thus, by specifying the text co-occurrence relationship, specific metadata contained in the ontology helps disambiguate the correct person, depending on the topics mentioned in the text.

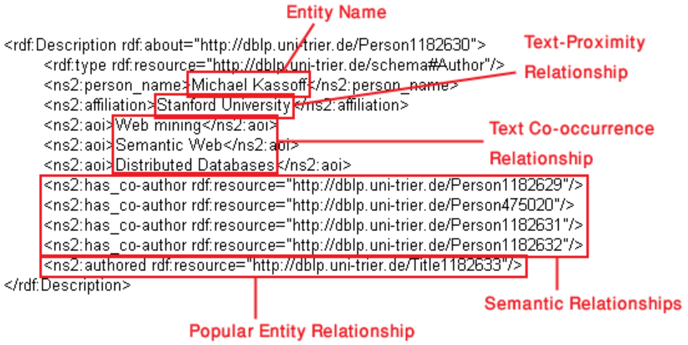
It is important to mention that this co-occurrence relationship is applicable only on well focused content. That is, if a document contains multiple DBWorld postings then its content could bring 'noise' and negatively impact the results of the disambiguation process. In such cases, it may be necessary to perform a text-segmentation process [9] to separate and deal with specific subparts of a document.

3.4 Popular Entities

The intuition behind using popular entities is to bias the right entity to be the one having more occurrences of 'popular' relationships (specified in advance). For example, researchers listed in Program Committees of DBWorld posts typically have a high number of publications. An 'author' relationship specified as popular can bias the candidate entities with many publications to be the right entity. For example, the abbreviated name "A. Joshi" matches up to 20 entities in DBLP but only a couple of such researchers have more than 70 papers published. The usage of this type of relationship for entity-disambiguation would depend on whether it is applicable for a given domain.

3.5 Semantic Relationships

Semantic relationships are intended to consider relationships that go beyond metadata which consists of literal values, such as syntactical matching of peoples' names [5]. For example, researchers are related to other researchers by means of their collaboration network. Researchers are also closely related to their co-authors and other authors through complex relationships. In DBWorld posts, it is common that the persons listed have relationships among themselves within a list of accepted papers and/or program committee members of a conference. Thus, the semantic relationship helps with determining the correct entity being referred in the text.



```
<rdf:Description rdf:about="http://dblp.uni-trier.de/Person1182630">
      <rdf:type rdf:resource="http://dblp.uni-trier.de/schema#Author"/>
      <ns2:person_name>Michael Kassoff</ns2:person_name>
      <ns2:affiliation>Stanford University</ns2:affiliation>
      <ns2:aoi>Web mining</ns2:aoi>
      <ns2:aoi>Semantic Web</ns2:aoi>
      <ns2:aoi>Distributed Databases</ns2:aoi>
      <ns2:has_co-author rdf:resource="http://dblp.uni-trier.de/Person1182629"/>
      <ns2:has_co-author rdf:resource="http://dblp.uni-trier.de/Person475020"/>
      <ns2:has_co-author rdf:resource="http://dblp.uni-trier.de/Person1182631"/>
      <ns2:has_co-author rdf:resource="http://dblp.uni-trier.de/Person1182632"/>
      <ns2:authored rdf:resource="http://dblp.uni-trier.de/Title1182633"/>
</rdf:Description>
```

Entity Name
Text-Proximity Relationship
Text Co-occurrence Relationship
Semantic Relationships
Popular Entity Relationship

Fig. 4. Sample RDF object

In Figure 4, we present a part of our DBLP RDF file, which is an input to the system for entity disambiguation. In this example, the entity's name is "Michael Kassoff" who is affiliated with "Stanford University" and has authored one paper. The author has three areas of interest and is related to four other authors via semantic relationships described above (e.g., has_co-author).

4 Algorithm

In this section, we describe our method for disambiguating entities in unstructured text. Figure 5 explains the steps of our method using pseudocode. The general idea is to spot entity names in text and then assign each potential match a confidence score. The confidence score for each ambiguous entity is adjusted based on whether existing information of the entity from the ontology matches accordingly to the relationship types found in the ontology as explained in the previous section. Throughout this paper, we will use *cf* to represent the initial confidence score, *acf* to represent the initial, abbreviated confidence score, *pr* to represent proximity score, *co* to represent text co-occurrence score, *sr* to represent the semantic relationship score and *pe* to represent the popular entity score. These variables are adjustable to capture the relative importance of each factor in the disambiguation process.

```
Algorithm Disambiguation( ) {
  for (each entity in ontology) {
    if (entity found in document) {
      create 'candidate entity'
      $\mathcal{C}_S$ for 'candidate entity' ← $cf$ / (entities in ontology)
    }
  }
  for (each 'candidate entity') {
    search for 'candidate entity''s text proximity relationship
    if (text proximity relationship found near 'candidate entity'){
      $\mathcal{C}_S$ for 'candidate entity' ← $\mathcal{C}_S$ for 'candidate entity' + $pr$
    }
    search for 'candidate entity''s text co-occurrence relationship
    if (text co-occurrence relationship found) {
      $\mathcal{C}_S$ for 'candidate entity' ← $\mathcal{C}_S$ for 'candidate entity' + $co$
    }
    if (ten or more popular entity relationships exist){
    {
       $\mathcal{C}_S$ for 'candidate entity' ← $\mathcal{C}_S$ for 'candidate entity' + $pe$
    }
  }
  iterate ← false
  while (iterate == true) {
    iterate ← true
    for (each 'candidate entity') {
      search  for semantic relationships in the ontology to other 'candidate entities'
      for (each relation found that has not been seen AND
           target entity $\mathcal{C}_S$ is above 'threshold') {
        $\mathcal{C}_S$ for 'candidate entity' ← $\mathcal{C}_S$ for 'candidate entity' + $sr$
        mark relation as seen
        if ('candidate entity' score has risen above 'threshold') {
          iterate ← false
}}}}}
```

Fig. 5. Algorithm pseudocode

4.1 Spotting Entity Names

The first step in our algorithm consists of spotting (within a document) the *names* of the entities to be disambiguated (see Section 3.1). The system only looks for entity-names of the ontology. Each entity name found in the document is a potential match for one or more entities in the populated ontology. Each of the entities of the ontology that matches a name becomes a *candidate entity*. A confidence score is initially assigned to each candidate entity depending on how many of them match the same name. The formula for assigning this confidence score ($\mathcal{C}_S$) is as follows.

$$e_S = \frac{cf}{\text{Number of entities with the same label}} \quad (1)$$

Techniques for spotting person names can be as simple as regular expressions that find anything that looks like a person name (e.g., two words having their first letter capitalized). We did not choose this type of techniques to avoid spotting irrelevant information, which would have had to be filtered out later. Our technique for spotting simply uses the known names of entities from the ontology and looks for them in the text (we were not very concerned with time-efficiency of this step in our prototype implementation). In addition to spotting based on name, this step also looks for abbreviated names, such as “A. Joshi”. This type of entities gets a e_S that is initialized differently to reflect the fact that many more entities from the ontology can syntactically match to the same name. The formula for assigning this confidence score in this case is as follows.

$$e_S = \frac{acf}{\text{Number of related entities in the ontology}} \quad (2)$$

The consideration for abbreviated names is a feature that can be turned on or off. We found that it is suitable for use with peoples' names yet we did not explore further considerations such as canonical names (i.e., Tim and Timothy) and other techniques for name matching [5, 13, 19].

4.2 Spotting Literal Values of Text-Proximity Relationships

The second step of our algorithm consists of spotting literal values based on *text-proximity* relationships (see Section 3.2). In order to narrow down the search for such literals, only the candidate entities found in the previous step are considered when determining literal values of text-proximity relationships to be spotted. By checking the ontology, it is then possible to determine whether a candidate entity appears near one of the spotted literal values based on text-proximity relationships, such as a known affiliation of a person appearing within a predefined window of the person name. We argue that this type of evidence is a strong indication that it might be the right entity. Hence, the confidence-score of an entity is increased substantially. Figure 2 shows an example where the affiliation is a highly relevant hint for the disambiguation of the candidate entity “Michael Kassoff.”

4.3 Spotting Literal Values of Text Co-occurrence Relationships

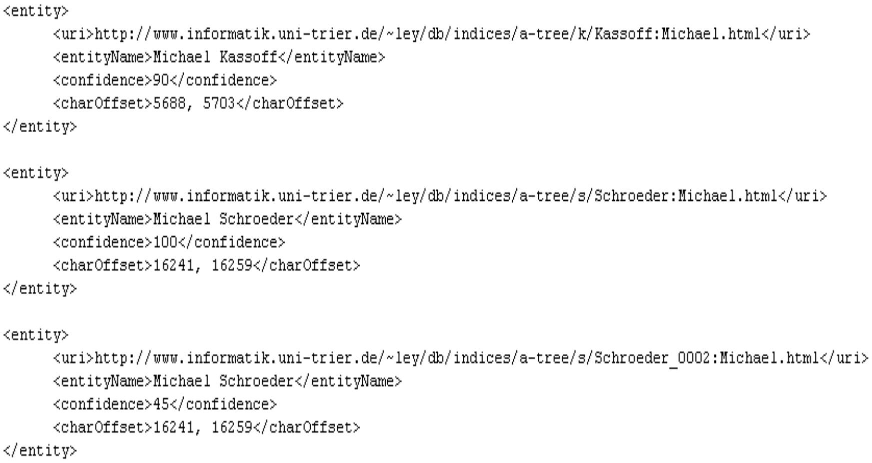
This step consists of spotting literal values based on *text co-occurrence* relationships (see Section 3.3). For every candidate entity, if one of its literal values considering text co-occurrence relationships is found within the document, its confidence score is increased. In our DBLP dataset, this step finds literal values appearing in the document based on the relationship ‘aoi’ which contains areas of interest of a researcher. For example, in Figure 3 “Web mining” and “Semantic Web” are spotted as *areas of interest* that match those of candidate entities. Thus, any candidate entity having such areas of interest receives an increase on its disambiguation e_S.

4.4 Using Popular Entities

The degree of popularity among the candidate entities is considered to adjust the e_S of candidate entities (see Section 3.4). The intention is to slightly increase the e_S for those entities that, according to the ontology, have many relationships that were predefined as popular (e.g. authored). In the scenario of DBWorld posts, this step slightly increases the score of candidate entities that have many publications as indicated in the ontology (as it is more likely that they would be listed in Program Committees). We acknowledge that this step may not be applicable in all domains. However, we found that it is a useful tie-breaker for candidate entities that have the same e_S.

4.5 Using Semantic Relationships

This step goes beyond just using literal values as evidence for disambiguating entities. The intuition is to use relationships to create a propagation or network effect that can increase the e_S of candidate entities based on *semantic* relationships (see Section 3. 5). In the scenario of disambiguating researchers in DBWorld posts, this step considers whether the candidate entities have co-authorship relationships and increases the e_S for the ones that do. Such e_S adjustments can only be done fairly by starting with the candidate entities having the highest score so far. Each candidate entity with a high score is analyzed through its semantic relationships in the ontology to increase the score of other candidate entities whenever they are connected through the ontology. On the other hand, it may not be necessary to perform this analysis on candidate entities with very low e_S. To deal with this issue, our algorithm uses a *threshold* e_S, which can be customized. Additionally, the process of adjusting e_S is repeated if at least one candidate entity gets its e_S increased over such threshold. Any such entity could then help boost the e_S of remaining candidate entities with low scores until no more adjustments to e_S take place. Thus, this step is iterative and always converges.



```
<entity>
      <uri>http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/k/Kassoff:Michael.html</uri>
      <entityName>Michael Kassoff</entityName>
      <confidence>90</confidence>
      <charOffset>5688, 5703</charOffset>
</entity>

<entity>
      <uri>http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/s/Schroeder:Michael.html</uri>
      <entityName>Michael Schroeder</entityName>
      <confidence>100</confidence>
      <charOffset>16241, 16259</charOffset>
</entity>

<entity>
      <uri>http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/s/Schroeder_0002:Michael.html</uri>
      <entityName>Michael Schroeder</entityName>
      <confidence>45</confidence>
      <charOffset>16241, 16259</charOffset>
</entity>
```

Fig. 6. Sample Output of Spotted Entities with their Disambiguation Score

4.6 Output

As shown in Figure 6, we have chosen to output our results in XML format because of its universally accepted syntax. For each entity found in the document and the ontology, we output its URI, name, confidence score and character offset. The URI of each entity represents the DBLP web page containing information regarding it. The name is the literal found in the documents and the character offset is the location of the entity within the document.

5. Evaluation

We chose to evaluate our method for entity disambiguation using a golden standard, which we created manually and we will refer to as *disambiguated dataset*. This dataset consists of 20 documents from DBWorld. For the purpose of having a representative dataset, the documents were chosen by first picking a random DBWorld announcement and the 19 next documents, as they were posted in chronological order. Each document was processed manually by inspecting peoples' names. For each person's name, we added a link to its corresponding DBLP web page, which we use in the ontology as the URI that uniquely identifies a researcher. Ideally, every DBWorld post would have a golden standard representation but this does not exist because it is extremely time consuming to create. By creating this disambiguated dataset, it is possible to evaluate our method's results and measure precision and recall.

We use a set *A* as the set of unique names identified using the disambiguated dataset and a set *B* as the set of entities found by our method. The intersection of these sets represents the set of entities correctly identified by our method. We measured precision as the proportion of correctly identified entities with regard to *B*. We measured recall as the proportion of correctly disambiguated entities with regard to *A*.

$$\mathscr{Precision} = \frac{sizeof(A\cap B)}{sizeof(B)} \tag{3}$$

$$\mathscr{Recall} = \frac{sizeof(A\cap B)}{sizeof(A)} \tag{4}$$

Our method computes the $\mathcal{C}_S$ of candidate entities using weights for the different disambiguation aspects in Section 4. These weights are part of input settings that allow fine tuning depending on the domain and importance of available relationships in a given ontology. We adjusted the settings so that an entity's affiliation and relations (co-authorship) to other researchers is considered far more valuable than the areas of interest of the researcher. Table 2 lists the assignments that produced the most accurate results when running our test data.

Within our golden standard set of documents, we were able to find 758 entities that have representations in our ontology. In the 20 documents of our disambiguated-set, only 17 person names were not represented in the DBLP ontology. These mainly consisted of local organizers and researchers listed in cross-disciplinary conferences.

Table 2. Values of Input Settings used in the Evaluation

Description	**Variable**	**Value**
charOffset		50
Text proximity relationships	*pr*	50
Text co-occurrence relationships	*co*	10
Popular entity score	*pe*	10
Semantic relationship	*sr*	20
Initial confidence score	*cf*	90
Initial abbreviated confidence score	*acf*	70
Threshold	*threshold*	90

When comparing the results of our method with the disambiguated-set, our method was able to find 620 entities. Only 18 of these were incorrectly disambiguated. We calculated the precision to be 97.1 percent and recall to be 79.6 percent. Table 3 is a summary of our results.

Table 3. Precision and Recall

Correct Disambiguation	**Found Entities**	**Total Entities**	**Precision**	**Recall**
602	620	758	97.1%	79.4%

Figure 7 illustrates the precision and recall evaluation on a per document basis. The document numbers coincide with our golden standard set available at http://lsdis.cs.uga.edu/~aleman/research/dbworlddis/. The precision is quite accurate in most cases and the recall varies from document to document.

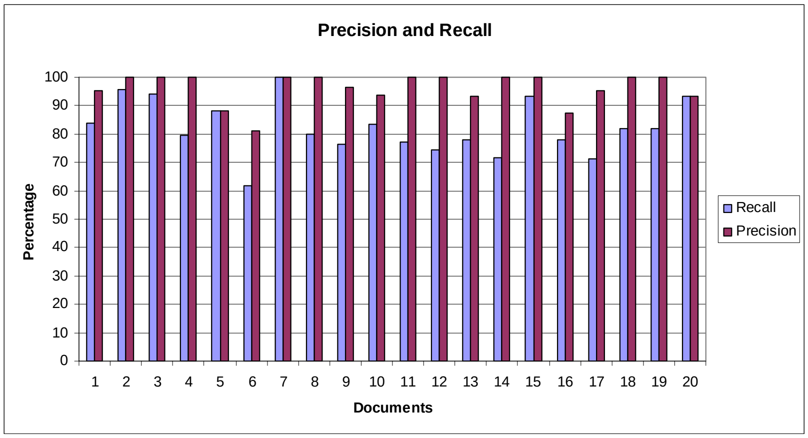


Fig. 7. Measures of Precision and Recall in a per-document basis

There are several situations where our method did not disambiguate the correct entity. This was mostly due to the ontology which, although largely populated, does not have complete coverage. For example, some of the authors within the ontology have only one relationship to a paper while some authors have a variety of relationships to papers, other authors, affiliation, etc. Because of this, it was not possible to precisely disambiguate some entities. Another error that is common is the situation where we find an entity's name that matches a portion of the name of another entity. We provide some safeguards against this as long as both of the candidate entities exist in the ontology, but the algorithm still misses in a few cases.

6. Related Work

Research on the problem of entity disambiguation has taken place using a variety of techniques. Some techniques only work on structured parts of a document. The applicability of disambiguating peoples' names is evident when finding citations within documents. Han et al [13] provides an assessment of several techniques used to disambiguate citations within a document. These methods use string similarity techniques and do not consider various candidate entities that may have the same name.

Our method differs from other approaches by a few important features. First, our method performs well on unstructured text. Second, by exploiting background knowledge in the form of a populated ontology, the process of spotting entities within the text is more focused and reduces the need for string similarity computations. Third, our method does not require any training data, as all of the data that is necessary for disambiguation is straightforward and provided in the ontology. Last but not least, our method exploits the capability provided by relationships among entities in the ontology to go beyond techniques traditionally based on syntactical matches.

The iterative step in our work is similar in spirit to a recent work on entity reconciliation [8]. In such an approach, the results of disambiguated entities are propagated to other ambiguous entities, which could then be reconciled based on recently reconciled entities. That method is part of a Personal Information Management system that works with a user's desktop environment to facilitate access and querying of a user's email address book, personal word documents, spreadsheets, etc. Thus, it makes use of predictable structures such as fields that contain known types of data (i.e., emails, dates and person names) whereas in our method we do not make any assumptions about the structure of the text. This is a key difference as the characteristics of the data to be disambiguated pose different challenges. Our method uses an ontology and runs on un-structured text, an approach that theirs does not consider.

Citation matching is a related problem aiming at deciding the right citation referring to a publication [11]. In our work, we do not assume the existence of citation information such as publication venue and date. However, we believe that our method is a significant step to the Identity Uncertainty problem [16] by automatically determining unique identifiers for person names with respect to a populated ontology.

KIM is an application that aims to be an automatic ontology population system that runs over text documents to provide content for the Semantic Web [17]. The KIM platform has many components that are unrelated to our work but within these components, there is an entity recognition portion. KIM disambiguates entities within a document by using a natural language processor and then attempts to index these entities. The evaluation of the KIM system is done by comparing the results to human-annotated corpora, much like our method of evaluation.

The SCORE system for management of semantic metadata (and data extraction) also contains a component for resolving ambiguities [18]. SCORE uses associations from a knowledgebase to determine the best match from candidate entities but detailed implementation is not available from this commercial system.

In ESpotter, named entities are recognized using a lexicon and/or atterns [20]. Ambiguities are resolved by using the URI of the webpage to determine the most likely domain of the term (probabilities are computed using hit count of search-engine results). The main difference with our work is our method uses only named entities within the domain of a specific populated ontology.

Finally, our approach is different to that of disambiguating word senses [2, 12, 15]. Instead, our focus is to disambiguate named entities such as peoples' names, which has recently gained attention for its applicability in Social Networks [3, 1]. Thus, instead of exploiting homonymy, synonymy, etc., our method works on relationships that real-world entities have such as affiliation of a researcher and his/her topics.

7. Conclusions

We proposed a new ontology-driven solution to the entity disambiguation problem in unstructured text. In particular, our method uses relationships between entities in the ontology to go beyond traditional syntactic-based disambiguation techniques. The output of our method consists of a list of spotted entity names, each with an entity disambiguation score e_S. We demonstrated the effectiveness of our approach through evaluations against a manually disambiguated document set containing over 700 entities. This evaluation was performed over DBWorld announcements using an ontology created from DBLP (consisting of over one million entities). The results of this evaluation lead us to claim that our method has successfully demonstrated its applicability to scenarios involving real-world data. To the best of our knowledge, this work is among the first which successfully uses a large, populated ontology for identifying entities in text without relying on the structure of the text.

In future work, we plan to integrate the results of entity disambiguation into a more robust platform such as UIMA [10]. The work we presented can be combined with other existing work so that the results may be more useful in certain scenarios. For example, the results of entity-disambiguation can be included within a document using initiatives such as Microformats (microformats.org) and RDFa (w3.org/TR/xhtml-rdfa-primer/).

Acknowledgments. This work is funded by NSF-ITR-IDM Award#0325464 (SemDIS: Discovering Complex Relationships in the Semantic Web).

References

1. Aleman-Meza, B., Nagarajan, M., Ramakrishnan, C., Ding, L., Kolari, P., Sheth, A., Arpinar, I. B., Joshi, A., Finin, T.: Semantic Analytics on Social Networks: Experiences in Addressing the Problem of Conflict of Interest Detection. *15th International World Wide Web Conference,* Edinburgh, Scotland (May 23-26, 2006)

2. Basili, R., Rocca, M. D., Pazienza, M. T.: Contextual Word Sense Tuning and Disambiguation. *Applied Artificial Intelligence*, 11(3) (1997) 235-262
3. Bekkerman, R., McCallum, A.: Disambiguating Web Appearances of People in a Social Network. *14th International World Wide Web Conference*, Chiba, Japan, (2005) 463-470
4. Berners-Lee, T., Fielding R., Masinter, L.: Uniform Resource Identifier (URI): Generic Syntax. *RFC 3986, IETF,* (2005)
5. Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P., Fienberg, S.: Adaptive Name Matching in Information Integration. *IEEE Intelligent Systems*, 18(5). (2003) 16-23
6. DBWorld. http://www.cs.wisc.edu/dbworld/ April 9, 2006
7. Dey, D., Sarkar, S., De, P.: A Distance-Based Approach to Entity Reconciliation in Heterogeneous Databases. *IEEE Transactions on Knowledge and Data Engineering*, 14(3) (May 2002) 567-582
8. Dong, X. L., Halevy, A., Madhaven, J.: Reference Reconciliation in Complex Information Spaces. *Proc. of SIGMOD*, Baltimore, MD. (2005)
9. Embley, D. W., Jiang, Y. S., Ng, Y.: Record-Boundary Discovery in Web Documents. *Proc. of SIGMOD*, Philadelphia, Pennsylvania (1999) 467-478
10. Ferrucci, D., Lally, A.: UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Natural Language Engineering*, 10(3-4) (2004) 327-348
11. Giles, C.L., Bollacker, K.D., Lawrence, S.: CiteSeer: An Automatic Citation Indexing System. *Proc. of the 3rd ACM International Conference on Digital Libraries*, Pittsburgh, PA, (June 23-26, 1998) 89-98
12. Gomes, P., Pereira, F. C., Paiva, P., Seco, N., Carreiro, P., Ferreira, J. L., Bento, C.: Noun Sense Disambiguation with WordNet for Software Design Retrieval. *Proc. of the 16th Conference of the Canadian Society for Computational Studies of Intelligence (AI 2003)*, Halifax, Canada (June 11-13, 2003) 537-543
13. Han, H., Giles, L., Zha, H., Li, C., Tsioutsiouliklis, K.: Two Supervised Learning Approaches for Name Disambiguation in Author Citations. *Proc. ACM/IEEE Joint Conf on Digital Libraries,* Tucson, Arizona (2004)
14. Ley, M.: The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives. *Proc. of the 9th International Symposium on String Processing and Information Retrieval,* Lisbon, Portugal (Sept. 2002) 1-10
15. Navigli, R., Velardi, P.: Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 27(7) (2005) 1075-1086
16. Pasula, H., Marthi, B., Milch, B., Russell, S. J., Shpitser, I.: Identity Uncertainty and Citation Matching, Neural Information Processing Systems. Vancouver, British Columbia (2002) 1401-1408
17. Popov, B., Kiryakov, A., Kirilov, A., Manov, D., Ognyanoff, D., Goranov, M.: KIM - Semantic Annotation Platform. *Proc. of the 2nd International Semantic Web Conference,* Sanibel Island, Florida (2003)
18. Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K., Warke, Y.: Managing Semantic Content for the Web, *IEEE Internet Computing*, 6(4), (2002) 80-87
19. Torvik, V. I., Weeber, M., Swanson, D. R., Smalheiser, N. R.: A Probabilistic Similarity Metric for Medline Records: A Model for Author Name Disambiguation. *Journal of the American Society for Information Science and Technology*, 56(2) (2005) 40-158
20. Zhu, J., Uren, V., Motta, E.: ESpotter: Adaptive Named Entity Recognition for Web Browsing, *Proc. of the 3rd Professional Knowledge Management Conference (WM2005)*, Kaiserslautern, Germany (2005)