UNIVERSITY
OF TRENTO

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.dit.unitn.it

COREFERENCE RESOLUTION ON RDF GRAPHS GENERATED FROM INFORMATION EXTRACTION: FIRST RESULTS

Mikalai Yatskevich, Christopher Welty and J. William Murdock

September 2006

Technical Report # DIT-06-057

Coreference resolution on RDF Graphs generated from Information Extraction: first results

Mikalai Yatskevich[1], Christopher Welty[2], J. William Murdock[2]

[1]Dept. of Information and Communication Technology
University of Trento,
38050 Povo, Trento, Italy
yatskevi@dit.unitn.it
[2]IBM Watson Research Center
19 Skyline Dr., Hawthorne, NY 10532
{welty, murdockj}@us.ibm.com

Abstract. In our research on the use of information extraction to help populate the semantic web, we have encountered significant obstacles to interoperability. One such obstacle is cross-document coreference resolution. In this paper we describe an effort to improve coreference resolution on RDF graphs generated by text analytics. In addition to driving knowledge-base population, our goal is to demonstrate that successfully combining semantic web and natural language processing technologies can offer advantages over either in isolation, and motivates overcoming the obstacles to interoperability. We present some early results that show improvement of coreference resolution using graph-matching algorithms over RDF.

Keywords: coreference resolution, information extraction, RDF, graph matching

1 Introduction

We are working on a project that is exploring the use of large-scale information extraction from text to address the "knowledge acquisition bottleneck" in populating large knowledge-bases. This is by no means a new idea, however our focus is less on theoretical properties of NLP or KR systems in general, and more on the realities of these technologies *today*, and how they can be used together. In particular, we have focused on state-of-the art text extraction components, many of which consistently rank in the top three at competitions such as ACE (Luo, et al, 2004) and TREC (Chu-Carroll, et al, 2005), that have been embedded in the open-source Unstructured Information Management Architecture (UIMA) (Ferrucci & Lally, 2004), and used to populate semantic-web knowledge-bases.

Populating the semantic web from text analysis is also not a particularly new idea; recent systems based on GATE (e.g. (Popov, et al 2004)) have been exploring the

production of large RDF repositories from text. In our project, however, we are specifically focused on the *nature of the data* produced by information extraction techniques, and its *suitability for reasoning.* Most systems that we have come across (see the related work section) do not perform reasoning (or perform at best the most simplistic reasoning) over the extracted knowledge stored in RDF, as the data is either too large or too imprecise.

In pursuing our goals, we have encountered five major obstacles to interoperability between today's technology in these areas, most notably the intolerance of KR&R systems to imprecision and low recall, the need for relation extraction, explanations, and scalability. In this paper, we focus on the fifth problem, which is perhaps the most important shortcoming in IE today for our purposes: the need for cross-document coreference.

Many of the problems we are investigating have theoretical solutions published in the literature, e.g. better NLP using deep parsing or more sophisticated reasoning in context logic. Neither information extraction nor semantic web technologies, in fact, represent the best theoretical work in their respective areas. We have found these technologies, however, to be widely available, supported by existing and emerging standards, and penetrating the IT industry. Our focus, again, is on actual systems that can populate and reason over the semantic web today, not theoretical results that do not have robust implementations. We believe by demonstrating empirically the combination of these technologies, we can begin to motivate more collaboration in theoretical work as well.

In this paper we will briefly discuss our general approach to generating OWL knowledge-bases from text, and describe our work in coreference analysis. In addition to our central goal of populating knowledge-bases from text, we also seek to demonstrate that combining KR and NLP technologies as they exist today offers advantages over either technology in isolation. We have shown in recent work (Welty and Murdock, 2006) that reasoning can be used to improve precision and recall of relation extraction, in this paper we show initial results on using the semantics of RDF graphs to improve coreference resolution, which itself is a critical requirement for populating knowledge-bases from text.

2 Related Work

Research on extraction of formal knowledge from text (e.g., Dill, Eiron, et al. 2003) typically assumes that text analytics are written for the ontology that the knowledge should be encoded in. Building extraction directly on formal ontologies is particularly valuable when the extraction is intended to construct or modify the original ontology (Maynard, Yankova, et al. 2005; Cimiano & Völker, 2005). However, there is a substantial cost to requiring text analytics to be consistent with formal ontology languages. There are many existing systems that extract entities and relations from text using informal ontologies that make minimal semantic commitments (e.g., Marsh, 1998; Byrd & Ravin, 1999; Liddy, 2000; Miller, Bratus, et al., 2001; Doddington, Mitchell, et al., 2004). These systems use these informal ontologies because those ontologies are relatively consistent with the ambiguous ways

concepts are expressed in human language and are well-suited for their intended applications (e.g., document search, content browsing). However, those ontologies are not well-suited to applications that require complex inference.

Work on so-called *ontology-based* information extraction, such as compete in the ACE program, (e.g. (Cunningham, 2005), (Bontcheva, 2004)) and other approaches from the semantic-web like (Maynard, 2005), (Maynard, et al, 2005), and (Popov, et al, 2004), focus on directly populating small ontologies that have a rich and well-thought out semantics, but very little if any formally specified semantics (e.g. using axioms). The ontologies are extensively described in English, and the results are apparently used mainly for evaluation and search, not to enable reasoning. Our work differs in that we provide an explicit knowledge integration step that allows us to populate fully axiomatized ontologies from information extraction.

Our emphasis actually makes our work similar to work in semantic integration or schema matching (e.g., Milo & Zohar, 1998; Noy & Musen, 2001), which typically focuses on finding very simple (e.g., one-to-one) mappings among terms in ontologies. However, state of the art matching systems (see Giunchiglia, et al., 2004, Ehrig & Staab, 2004) for example and (Shvaiko & Euzenat, 2005) for recent survey) are focused on determining the correspondences holding among ontology classes but not the instances.

The existing natural language approaches to coreference resolution have used decision trees (McCarthy and Lehnert, 1995; Ng and Cardie, 2002); SVMs (Zelenko *et al.*, 2003); maximum entropy classifiers (Morton, 1997); generative probabilistic models (Ge *et al.*, 1998), (Li *et al.*, 2005); KL-divergence, agglomerative and incremental vector spaces (Gooi and Allan, 2004) and bi-gram co-occurrences (Pendersen *et al.*, 2005) in order to learn the pairwise distance measure. But none of them have used ontology level information to determine if or where a coreferent merge should occur.

Personal name disambiguation and coreference problems received a considerable attention in the last years. See (Mann and Yarowsky, 2003), (Niu *et al.*, 2004) for recent examples. Differently from these approaches we do not restrict ourselves to personal names allowing coreference resolution of any entity in the predefined ontology.

We use in our work components implemented within the Unstructured Information Management Architecture (UIMA). UIMA is an open-source middleware platform for integrating components that analyze unstructured sources such as text documents. UIMA-based systems define “type systems” (i.e., ontologies with extremely limited semantic commitments) to specify the kinds of information that they manipulate (Götz & Suhre, 2004). UIMA type systems include no more than a single-inheritance type/subtype hierarchy, thus to do substantive reasoning over the results of UIMA-based extraction, one needs to convert results into a more expressive representation.

3 Generating RDF from Text

The context of our application deserves some attention, as our results are somewhat dependent on the assumptions that arise from it. Our OWL ontologies are small,

consisting of no more than 100 classes and 100 object properties, which makes sense if they are to be populated from text analysis, as typical information extraction ontologies are extremely small.

Analytics are available in reusable components that can be embedded in frameworks like UIMA, in which they are composed into larger aggregate analysis engines. The individual components assign labels (or *annotations*) that carry some semantics to regions of the data, as well as coreference within documents. The field is well represented by the ACE program (Doddington, et al, 2004), participants in which produce annotations for entities (Person, Organization, etc.), relations (*partOf*, *citizenOf*, etc.), and coreference analysis. The components we use overlap to varying degrees in the types of entities and relations they discover, and in the cases of overlap, need to have their results combined. While this has in general been shown to improve overall precision and recall, it does create interesting anomalies in the results (which we will discuss below). The individual analytic components we treat as black boxes, their operation is for the most part functional (producing the same output for the same input) and unchangeable – while we can ask for bug fixes, we take it as given that these analytics will produce errors and try to deal with it.

In addition to this *document-level processing*, our system also performs analysis at the *collection level*, and the most important such analysis is cross-document coreference analysis – the identification of individual entities that are mentioned (and annotated) in multiple places. Again, many of our document-level components produce coreference analysis within documents, but connecting these results across the entire corpus clearly requires processing that can collect information from all the documents, and thus will typically scale at a polynomial rate. In our experience, the most critical properties of coreference are recognition of aliases and nicknames, common spelling variations of names (especially in other languages), common diminutives, abbreviations, etc. This is a wide-open research area that requires significant attention, and as discussed previously, coreference is a critical part of producing knowledge-bases from text.

After collection-level processing, we have data in a form that can be used to populate an OWL knowledge-base. We generate these knowledge-bases through a process called *knowledge integration*, realized in our Knowledge Integration and Transformation Engine (KITE) (Murdock and Welty, 2006). This gives us an RDF graph whose nodes are the entities resulting from collection-level processing, and the arcs are the relations discovered between them.

4 Improving coreference resolution

At the present time, coreference is a task in the text processing pipeline that consists of in-document coreference (including anaphora) and cross-document coreference that produces a graph of entities and relations which are mapped into RDF. A more semantic coreference algorithm is then run on the RDF.

4.1 Text-based coreference

Our within-document coreference combines a statistical component and a rule-based component that were developed separately for unrelated projects and integrated via UIMA. A new component, which was developed for this research, is used to combine the results of the within-document coreference resolution systems. That component uses the results of both within-document coreference resolvers to determine sets of spans that are coreferential and then determines the type of the referent using votes from a variety of statistical and rule-based named entity recognizers. The final results of this component are typed sets of spans within a single document; each of these sets is asserted to refer to a single entity or relation instance.

Cross-document coreference resolution takes these sets as input and determines which of them are coreferential across multiple documents. For example, it might receive spans whose texts are "Britain" and "the UK" in one document and spans whose texts are "Great Britain" and "that country" in another document and conclude that both sets are referring to a single entity.

In our system, cross-document coreference resolution is accomplished using a family of components with different applicability conditions. For example, a nation coreference resolver uses a table of alternative names of nations to determine which ones are coreferential. A date/time coreference resolver uses a normalized form of the dates and times (e.g., "Dec 8, 2003" and "12/8/03" are both normalized to "2003-12-8") and performs exact string matching over those normalized forms. A generic cross-document entity coreference resolver handles those types for which we do not have more specialized coreference resolution. It uses string matching plus expansion of abbreviations and gives preference to forms that have been identified as proper names.

Our within document coreference components have participated in the ACE competition and rank in the top five. There is no established (i.e., agreed upon by an active community) corpus or metric for evaluating cross-document coreference that we are aware of. This makes it extremely difficult to effectively compare our results to other approaches. Our experience with text-based coreference components is that their performance varies drastically from corpus to corpus, thus comparing two techniques based on numerical measures like precision and recall is meaningless.

The output of cross-document coreference resolution is typed entity and relation instances, some of which may be referred to in multiple documents. These results are then mapped (via the knowledge integration process) into an RDF graph that populates a specified OWL ontology.

4.2 Knowledge-based coreference post-processing algorithm

The key underlying idea of our approach is to exploit the knowledge implicitly and explicitly encoded in the RDF graphs to revise the results of text-based coreference resolution. In particular we exploit a graph matcher (Melnik et. al, 2002) in order to obtain a set of correspondences holding between the instance nodes of the RDF graph. The technique is based on the key intuition that similar instances tend to participate in similar sets of relations.

The process is structured as follows:

- *Step 1*: Instance matching
- *Step 2*: Graph matching
- *Step 3*: Result filtering

In step 1 the set of similarity coefficients holding between instances of the graph is produced. In this step we do not consider the structure of the graph (i.e., the relations in which the given instance participates) but exploit a special purpose instance matcher which considers only the mentions (i.e. the spans of text) attached to the instances. The instance matcher takes as an input 2 instances (I_1 and I_2) and produces a numerical similarity coefficient [0,1] (Sim_I). The matcher computes Sim_I as follows:

1. The instance with the smallest number of mentions I_1 is selected.
2. Each mention of I_1 is matched against each mention of I_2 exploiting a string distance matcher (Levenshtein, 1966). The mention of I_1 is called connected if its string similarity with at least one mention of I_2 is more than the predefined threshold.
3. Instance similarity is calculated as follows $Sim_I = C_M/N_M$, where C_M is the number of connected mentions in I_1 and N_M is a number of mentions in I_1.

Figure 1 illustrates the instance matcher algorithm. As from the figure two out of three mentions of I_1 (*George Bush* and *George*) are connected with the mentions of I_2 (i.e., their string similarity exceed the given threshold taken as 0.4 in the figure). Therefore, for the case depicted in the figure C_M=2, N_M=3 and Sim_I =2/3=0.66.

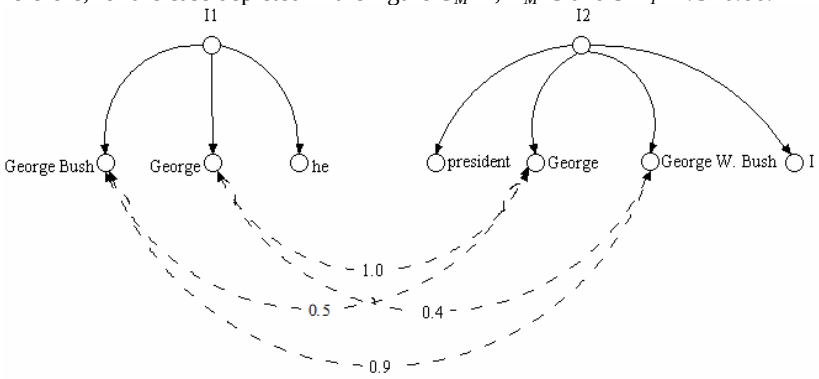

Fig. 1. Two instances. Solid lines designate mention of relation. Dashed lines are drawn between connected mentions.

In step 2 the set of similarities produced on the previous step is refined taking into account the graph structure. In order to perform this we exploit the *Similarity Flooding (SF)* graph matching algorithm (Melnik et. al, 2002). Similarity Flooding takes two labeled graphs and a set of initial (seed) similarities as an input and produces a set of similarity coefficients [0,1] holding between the nodes of both graphs. The basic concept behind the SF algorithm is that similarity spreads from similar nodes to their adjacent neighbors through propagation coefficients. From iteration to iteration, the spreading depth and a similarity measure increase until a fix-point is reached. In our case the RDF graph is matched against itself, and the set of similarities obtained on step 1 is used as seeds.

In step 3 the set of correspondences holding between instances in the RDF graph is produced by filtering the results of step 2, and instances are merged accordingly. The goal of step 3 is to filter the correspondences that violate the constraints of the domain ontology. Instances are merged only in the case when either:

1. They belong to the same class in the domain ontology.
2. Their string similarity is higher than a predefined (and very high (0.9)) threshold.

The first case allows enforcing of domain constraints (e.g., *Person* will never be merged with a *City*). The second case allows us to deal with errors in class annotation (i.e., with the fact that two very similar instances can be erroneously classified under two different classes). In our experiments we also exploited two optional filtering rules:

- *GEO:* If two instance labels are geographical names and their string similarity is less then a given threshold (0.9), the instances are different.
- *Person:* If two instance labels are Person names and their string similarity is less then a given threshold (0.9), the instances are different.

4.3 Empirical evaluation

We have implemented the algorithm described in the previous section and evaluated it on three real world datasets. The dataset properties are presented on Table 1. All three data sets are derived from subsets of the same corpus. The corpus is a set of unclassified news abstracts/summaries gathered by the Center for Nonproliferation Studies. The average size of documents in this corpus is approximately 2 kilobytes. The *D1* and *D2* datasets contain all of the triples for the subcorpora that were obtained by the combination of UIMA-based extraction & coreference followed by KITE-based mapping. The *D2 pruned* dataset uses the same subcorpus as the D2 dataset, but has also had the triples that violate the constraints of the ontology pruned out (Welty and Murdock, 2006) prior to the use of knowledge-based coreference post-processing.

Table 1. Dataset statistics.

	# of documents	# of RDF triples	# of filtered RDF triples	# of filtered instances
D1	50	2973	461	230
D2 pruned	378	22905	2403	1146
D2	378	23783	2676	1218

The number of documents in each subcorpus is presented in the second column. The third column presents the number of RDF triples extracted from the documents. As in (Welty and Murdock, 2006), most instances are disconnected (i.e., participate only in rdf:label triples and not connected by any relation with the other instances in the RDF graph). We have removed such instances from consideration. The number of triples and instances left after removal are listed in the fourth and fifth columns.

We have manually evaluated the merges suggested by our knowledge-based coreference resolution post-processing algorithm. The graph matching algorithm outputs a ranked list of pairwise correspondences. This allows us to define the cut-off

threshold and to consider only the correspondences with similarity above the defined threshold. Precision (i.e., the ratio of correctly suggested instance merges to total number of suggested instance merges) of the algorithm at various cut off threshold levels is presented in Figures 2a-c. We cannot calculate recall since we do not know the total number of correct instance merges. However, Figures 2a-c are similar to precision/recall curves. They present the change in precision for increasing number of merges. Points on the figures stand for various cutoff thresholds. Notice also that GEO and Person heuristics significantly improved precision on all datasets. Figure 2d) presents time performance of the algorithm depending on the number of triples. This data suggests that the execution time grows roughly linearly in respect with number of triples under consideration. The most computationally expensive part of the process is the instance matching, which means that the time performance of the system is limited by the performance of its string matching subroutine.

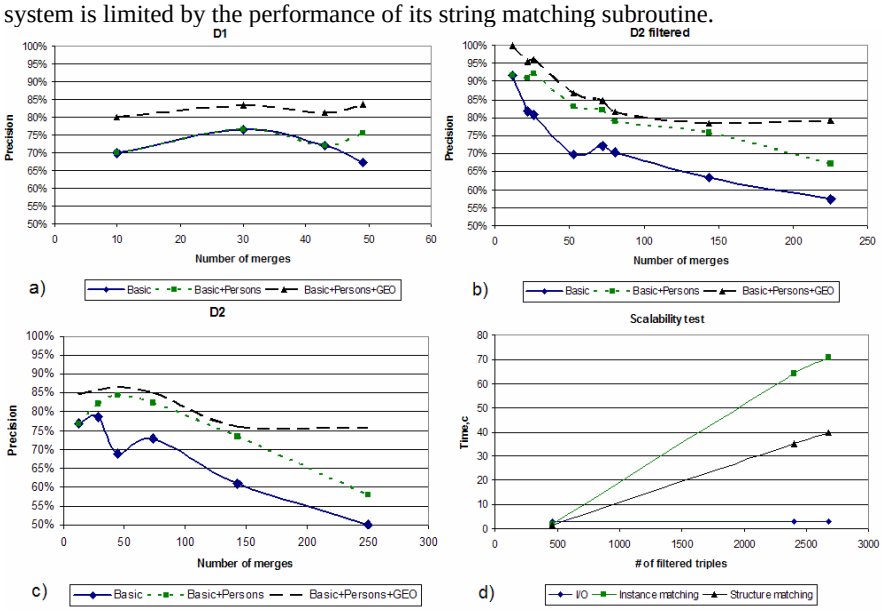

Fig. 2. Precision at various cut off thresholds for: a) D1 dataset; b) D2 pruned dataset; c) D2 dataset. d) Time spent on I/O operations, instance and structure matching for datasets of various sizes.

5 Conclusions

To reason about the contents of a set of documents, it is essential to recognize when documents are referring the same real-world entities. Text-based strategies for detecting cross-document coreference are able to accomplish this task to a limited degree. However, such strategies are not nearly reliable enough for current state-of-the-art knowledge representation & reasoning systems, which generally have low

tolerance for inaccurate knowledge-bases. We have shown how semantically-rich ontological information can be used over an extracted knowledge-base to enhance the quality of the coreference resolution in that knowledge-base. Reducing the frequency of coreference errors brings us closer to bridging the gap between extraction technology and reasoning technology.

In addition, we have shown that combining information extraction and semantic web technologies can offer improvements over either technology in isolation. Although these initial results are quite modest, we feel they are the "tip of the iceberg" and that further research on realistic interoperation between these kinds of systems can be very fruitful.

References

A. Bagga and B. Baldwin. Algorithms for Scoring Coreference Chains. In The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference, May 1998.

K. Bontcheva. 2004. Open-source Tools for Creation, Maintenance, and Storage of Lexical Resources for Language Generation from Ontologies. Fourth International Conference on Language Resources and Evaluation (LREC'2004). Lisbon, Portugal. 2004.

R. Byrd & Y. Ravin. 1999. Identifying and Extracting Relations in Text. 4th International Conference on Applications of Natural Language to Information Systems (NLDB). Klagenfurt, Austria.

J. Chu-Carroll, K. Czuba, P. Duboue, and J. Prager. 2005. IBM's PIQUANT II in TREC2005. The Fourteenth Text REtrieval Conference (TREC 2005) Proceedings.

P. Cimiano, J. Völker. 2005. Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery. 10th International Conference on Applications of Natural Language to Information Systems (NLDB). Alicante, Spain.

H. Cunningham,. 2005. Automatic Information Extraction. Encyclopedia of Language and Linguistics, 2cnd ed. Elsevier.

S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin, & J. Y. Zien. 2003. SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. 12th International World Wide Web Conference (WWW), Budapest, Hungary.

G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, & R. Weischedel. 2004. Automatic Content Extraction (ACE) program - task definitions and performance measures. Fourth International Conference on Language Resources and Evaluation (LREC).

M. Ehrig and S. Staab. QOM: Quick ontology mapping. In Proceedings of the International Semantic Web Conference (ISWC), pages 683–697, 2004.

D. Ferrucci & A. Lally. 2004. UIMA: an architectural approach to unstructured information processing in the corporate research environment. Natural Language Engineering 10 (3/4): 327–348.

N. Ge, J. Hale, and E. Charniak. A statistical approach to anaphora resolution. In Proceedings of the Sixth Workshop on Very Large Corpora, pages 161–171, 1998.

F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-Match: an algorithm and an implementation of semantic matching. In Proceedings of ESWS, pages 61–75, 2004.

T. Götz & O. Suhre. 2004. Design and implementation of the UIMA Common Analysis System. IBM Systems Journal 43 (3): 476-489.

C. H. Gooi, J. Allan: Cross-Document Coreference on a Large Scale Corpus. HLT-NAACL 2004: 9-16

V. I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Soviet Physics Doklady, 10(8):707-710, 1966.
X. Li, P. Morie, D. Roth: Semantic Integration in Text: From Ambiguous Names to Identifiable Entities. AI Magazine 26(1): 45-58
E. D. Liddy. 2000. Text Mining. Bulletin of American Society for Information Science & Technology.
X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla, S. Roukos: A Mention-Synchronous Coreference Resolution Algorithm Based On the Bell Tree. ACL 2004: 135-142.
G. S. Mann and D. Yarowsky, Unsupervised Personal Name Disambiguation. In: Proceedings of CoNLL-2003 , Edmonton, Canada, 2003, pp. 33-40.
E. Marsh. 1998. TIPSTER information extraction evaluation: the MUC-7 workshop.
D. Maynard. 2005. Benchmarking ontology-based annotation tools for the Semantic Web. UK e-Science Programme All Hands Meeting (AHM2005) Workshop "Text Mining, e-Research and Grid-enabled Language Technology", Nottingham, UK, 2005.
D. Maynard, M. Yankova, A. Kourakis, and A. Kokossis. 2005. Ontology-based information extraction for market monitoring and technology watch. ESWC Workshop “End User Apects of the Semantic Web,” Heraklion, Crete, May, 2005.
J. F. McCarthy and W. G. Lehnert. Using decision trees for coreference resolution. In IJCAI, pages 1050–1055, 1995.
S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm. In Proceedings of ICDE, pages 117–128, 2002.
S. Miller, S. Bratus, L. Ramshaw, R. Weischedel, A. Zamanian. 2001. FactBrowser demonstration. *First international conference on Human language technology research* HLT '01.
T. Milo, S. Zohar. 1998. Using Schema Matching to Simplify Heterogeneous Data Translation. VLDB 98, August 1998.
T. Morton. Coreference for NLP applications. In Proceedings ACL, 1997.
J. W. Murdock & C. Welty. 2006. Obtaining Formal Knowledge from Informal Text Analysis. IBM Research Report RC23961.
J. W. Murdock, D. McGuinness, P. Pinheiro da Silva, C. Welty, and D. Ferrucci1. 2006. Explaining Conclusions from Diverse Knowledge Sources. In Proceedings of ISWC-06.
V. Ng and C. Cardie. Improving machine learning approaches to coreference resolution. In Fortieth Anniversary Meeting of the Association for Computational Linguistics (ACL-02), 2002.
C. Niu, W. Li, R. K. Srihari: Weakly Supervised Learning for Cross-document Person Name Disambiguation Supported by Information Extraction. ACL 2004: 597-604
N. F. Noy & M. A. Musen. 2001. Anchor-PROMPT: Using Non-Local Context for Semantic Matching. Workshop on Ontologies and Information Sharing at the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-2001), Seattle, WA.
T. Pedersen, A. Purandare, A. Kulkarni: Name Discrimination by Clustering Similar Contexts. CICLing 2005: 226-237
B. Popov, A. Kiryakov, D. Ognyanoff, D. Manov, A. Kirilov. 2004. KIM - A Semantic Platform for Information Extraction and Retrieval. Journal of Natural Language Engineering, 10(3-4): 375-392.
P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. Journal on Data Semantics, IV, 2005.
C. Welty and J. W. Murdock. 2006. Towards Knowledge Acquisition from Information Extraction. In Proceedings of ISWC-06.
D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. Journal of Machine Learning Research (submitted), 2003.