

Automatic Entity Disambiguation: Benefits to NER, Relation Extraction, Link Analysis, and Inference

Matthias Blume

HNC Software LLC (a wholly-owned subsidiary of Fair Isaac Corporation)

3661 Valley Centre Drive

San Diego, CA 92130, USA

MatthiasBlume@FairIsaac.com

Keywords: Information Extraction and Link Analysis, Novel Intelligence from Massive Data, HUMINT, OSINT

Abstract

Entity disambiguation resolves the many-to-many correspondence between *mentions* of entities in text and unique real-world entities. Entity disambiguation can bring to bear global (corpus-level) statistics to improve the performance of named entity recognition systems. More importantly, intelligence analysts are keenly interested in relationships between real-world entities. Entity disambiguation makes possible additional types of relation assertions and affects relation extraction performance assessment. Finally, link analysis and inference inherently operate at the level of entities, not text strings. Thus, entity disambiguation is a prerequisite to carrying out these higher-level operations on information extracted from plain text. This paper describes Fair Isaac's automatic entity disambiguation capability and its performance.

1 Introduction

Spoken and written text consists of characters, words, names, sentences, documents, conversations, and so on, but the world that the text describes consists of distinct objects and events. Intelligence analysts have access to an enormous amount of text, but are ultimately interested in actual persons and organizations that interact in the real world. Entity disambiguation (sometimes also referred to as *entity tracking*) is the process of determining which names, words, or phrases in text correspond to distinct persons, organizations, locations, or other entities. This determination is absolutely essential for reasoning, inference, and the examination of social network structures based on information derived from text.

We use the term *entity* to mean an object or set of objects in the world. A *mention* is a reference to an entity such as a word or phrase in a document. Entities may be referenced by their name, indicated by a common noun or noun phrase, or represented by a pronoun. Mentions may aggregate with

other mentions that refer to the same specific real-world object, and, taken together, the aggregated mentions model an entity. These corpus-wide aggregated models of entities are of primary importance to the analyst, while the individual mentions of an entity are still of secondary importance (Mitchell et al. 2004).

Entity disambiguation inherently involves resolving many-to-many relationships. Multiple distinct strings, such as “Abdul Khan”, “Dr. Khan”, and “Abd al-Qadir Khan”, may refer to the same entity. Simultaneously, multiple identical mentions refer to distinct entities. For example, literally tens of thousands of men share the name “Abdul Khan”.

Consider the following sentences from a corpus of news text:

- “Young pacer Yasir Arafat chipped in with two late wickets to finish with two for 36 in 18 overs.”
- “General Musharraf also apprised Yasir Arafat of the outcome of his negotiations with the Indian Prime Minister Atal Behari Vajpayee at Agra.”
- “Palestinians demonstrated in Gaza City Sunday in support of Palestinian leader Yasser Arafat.”
- “Makkah has announced that the Arafat Gathering (9 Zul-Hajj) will be on Sat 31 Jan 2004.”

These can be confusing even to a human reader. The first underlined name refers to a Pakistani cricket player, the next two refer to the late Palestinian leader, and the last refers to a place near Mecca, Saudi Arabia. The job of the entity disambiguation system is to automatically assign these four mentions to three distinct entities, correctly grouping only the 2nd mention of “Yasir Arafat” with “Yasser Arafat”.

Fair Isaac's entity disambiguation software correctly resolves the above mentions in the context of the entire corpus. Our system is based largely on language-independent algorithms. Thus, it is applicable not only to unstructured text from arbitrary human languages, but also to semi-structured data such as citation databases and the disambiguation of named entities mentioned in wire transfer transaction records for the purpose of detecting money-

laundering activity. The system utilizes multiple types of context as evidence for determining whether two mentions correspond to the same entity and it automatically learns the weight of evidence of each context item via corpus statistics. It utilizes multiple search keys to efficiently find pairs of mentions that correspond to the same entity while skipping billions of unnecessary comparisons, yielding a system with very high throughput that can be applied to truly massive data.

The remaining sections of this paper describe Fair Isaac’s automatic entity disambiguation methodology, report on the performance of the system on a corpus of Pakistani news text, and mention implications of high-quality entity disambiguation for improving named entity recognition, relation extraction, link analysis, and inference.

2 Methodology

In unstructured text, each document provides a natural context for entity disambiguation. Within a document, two mentions of “Abdul Khan” probably do refer to the same person unless there is evidence to the contrary. Similarly, “NFC” and “National Fertilisers Corporation” probably refer to the same organization if they occur in the same document, barring other evidence. Thus, we first carry out within-document co-reference resolution, aggregating information about each entity mentioned in each document. We then use these entity attributes as features in determining which documents deal with the same entity.

2.1 Within-Document Disambiguation

When dealing with unstructured text, a named entity recognition (NER) system provides the input to the entity disambiguation. We currently use two NER systems in parallel. One is based on supervised training of hidden Markov models (Bikel 1997) followed by name list and rule-based post-processing. The other utilizes a set of engineered regular expressions. Both provide mention start and stop boundaries, entity type assertions, and confidence values. The two systems complement one another in that the former uses local context and provides higher coverage, whereas the latter is more accurate, especially for numeric types such as dates and monetary amounts.

In this environment, the first step in entity disambiguation is determining which mentions to deal with. The two NER systems sometimes mark overlapping text segments as being named entities and sometimes determine different entity types for the same mention. These are specific symptoms of a more general problem that would be present even if we used only a single NER system. In general, the performance of an NER system that uses only local context can be improved by using document-level (or corpus-level) information. For example, if the system determined that two mentions of “Christian Dior” refer to a company and one mention refers to a person, the performance could be improved by using a majority vote to re-categorize the person as a company. Similarly, a mention “Christian Dior Produced” could be identified as a segmentation error based on the information that “Christian Dior” was recognized as an en-

tity elsewhere, and a mention “Christian Dior” that was not recognized by the NER could be labeled as a company based on the evidence from the rest of this document.

Our algorithm resolves entity type discrepancies via majority vote over all identical strings that have been labeled, using name lists (person titles, given names, and family names as well as location tokens) and confidence values to resolve ties. It uses confidence values to resolve segmentation discrepancies, using string starting and ending positions to resolve ties. It then detects and repairs some NER segmentation errors (such as “Christian Dior Produced”) using document-level token sequence counts and word lists. Finally, it identifies additional mentions that were not recognized by the NER engine that are identical to recognized mentions, and labels those as entities. Only at this point have all entity mentions been identified (even though some disambiguation – inferring that “Christian Dior Produced” and “Christian Dior” refer to the same entity – has already been carried out).

The system next carries out entity type-specific parsing in order to extract entity attributes, generate standardized names (e.g. `p_abdul_khan_p` for “Dr. Abdul Q. Khan”), and populate the in-memory data structures (token hashes) that are used to perform the within-document entity disambiguation.

For example, in the case of person mentions, this *entity information extraction* process detects prefixes (titles) and suffixes (“Jr.”, “III”, “Esq.”), which are removed from the named itself and stored as attributes. The system uses a set of regular expressions to operate on token type sequences (as opposed to on the names themselves) and resolve ambiguities such as the usage of “Dean” in “Assistant Dean Smith” (title), “Dean Michael Smith” (title), “Col. Dean Smith” (given name), and “Michael Dean” (family name). Our list of given names includes gender probabilities and confidence levels, and this is used in conjunction with the title to infer the likely gender of the entity. The titles also provide evidence for entity attributes including job category (military, academic, religious, government, etc).

In the case of organizations, the system computes and stores likely acronyms for subsequent use in the disambiguation.

A list of entities is initialized as an empty list and the names are processed in order of longest to shortest within each entity type class. Each name is compared to all entities with which it may be compatible (based on the aforementioned token hash). The comparison is carried out via entity type-specific distance measures. For example, the person distance measure enforces gender consistency, deals with given name variants (“Mike” vs. “Michael”) using a given name variant list, and allows initials to match long forms of names. If a match is found, this name is assigned to the existing entity. If not, it is used to seed a new entity. If a name matches multiple entities, it is assigned to the one with the most recent mention (Hobbs 1978).

This within-document disambiguation process is not perfect, but it is close. Careful inspection indicates that less than one percent of the assignment operations are incorrect

– the error rate in this process is certainly lower than the named entity recognition error rate which underlies it. We err on the side of *not* merging entities rather than incorrectly merging entities. Specifically, there is no mechanism for merging names with spelling errors. Within a single document, it is very difficult to determine whether two distinct individuals named “Abdul Kahn” and “Abdul Kahan” really exist or whether a single instance of the latter string represents a spelling error. Looking at multiple documents provides additional statistics. Thus, the cross-document disambiguation process described in the next section will still merge some entities (including due to spelling errors) even within individual documents.

Some aspects of the within-document disambiguation process are language-specific. However, most of these have been separated as separate components (name lists and type sequence regular expressions) in the software architecture and can consequently be easily replaced for different languages.

These language-specific components may be viewed as pattern generators. If it is possible to create one observed string from another, the two mentions are legitimate variants of the same name. Table 1 lists several name variations and whether our algorithm considers each pair as possible variants of the same name. Clearly, a simple edit distance such as Levenshtein or string normalization such as Soundex could not produce the desired results. Automatically learning the desired transformation properties from a corpus of text (obviating the need for language-specific resources and rules) is work in progress.

American Broadcasting Corporation	Australian Broadcasting Corporation	N
ABC	Australian Broadcasting Corporation	Y
Dr. Ho	Dr. Hu	N
Dr. Ho	Professor Benjamin Xuhui Ho	Y
A. Smith	Michael A. Smith	N
M. Smith	Michael A. Smith	Y

Table 1. Name pairs (columns 1 and 2) and the desired outcome (column 3), indicating whether they should be considered possible variants of the same name.

2.2 Cross-Document Disambiguation

Within-document entity disambiguation appeared at first to be simple because of the small number of string pairs that need to be compared. Instead, it turned out to be complicated due to the numerous natural language conventions that humans intuitively use in generating and resolving these co-references. Cross-document disambiguation, in contrast, at first appeared difficult because of the large number of document-level entity pairs that need to be compared. In actuality, the facts that humans generally do not use linguistic constructs to generate cross-document co-references and that computers are good at carrying out millions of compari-

sons facilitate graceful and general (language-independent) automatic solutions.

Our cross-document entity disambiguation relies on one key insight: an entity can be distinguished by the company it keeps. If Abdul Khan 1 associates with different people and organizations at different locations than Abdul Khan 2, then he is probably a different person. Furthermore, if it is possible to compare two entities based on one type of context, it is possible to compare them based on *every* type of context. Utilizing additional context domains improves the entity disambiguation performance.

Within each domain, we require a finite set of context items. In the location, organization, and person domains, these are the standardized names derived in the entity information extraction phase of within-document disambiguation. We use the logarithm of the inverse name frequency (the number of standard names with which this context item appears), INF, as a weight indicating the salience of each context item. Co-occurrence with a common name provides less indication that two mentions correspond to the same entity than co-occurrence with an uncommon name. A sparse count vector is adequate for recording all of the items that co-occur with a particular entity.

We similarly create a sparse count vector of title tokens that occur with the entity and compute INF weights for the title tokens.

Finally, we create a word vector space in a completely unsupervised fashion (Caid and Oing 1997). Each document may be represented by a vector in the space. In order to avoid double-counting context features, we delete all named entity mentions from each document prior to computing its vector. An unsupervised clustering of some of the document vectors defines a segmentation of the space, and each document can then be uniquely assigned to a single segment. We again compute INF weights indicating the context’s salience, this time based on the number of standardized names that occur in documents falling into each segment.

We define a separate distance measure per context domain. It would be possible to use a single distance measure, such as dot product of the INF-weighted count vectors. However, the probabilities of occurring with multiple items within a single domain are conditional and not independent. The probability of two names both occurring with “Rawalpindi”, “Rawalpindi, Pakistan”, and “Islamabad” is not immensely different from the probability of occurring with only the least common of these three. We are able to discount the co-occurrence with multiple items as well as quantify an unexpected *lack* of shared co-occurrence by engineering each distance measure for each specific domain. The score produced by each distance measure may be loosely interpreted as the log of the likelihood of two *randomly generated* contexts sharing the observed degree of similarity.

In addition to the context-based distance measures, we utilize a lexical (string) distance measure. This is based on exactly the same transformations (and source code) as used to compare strings for intra-document entity disambiguation

plus the Soundex algorithm (Knuth 1998) to measure whether two name tokens sound the same. A large negative score indicates a great deal of similarity (log likelihood).

The process of cross-document entity disambiguation now boils down to repeatedly finding a pair of entities, comparing them (computing the sum of the above distance measures), and merging them if the score exceeds some threshold. Given N document-level entities in a corpus, comparing all pairs of entities would require $O(N^2)$ time (since each merge operation creates a new entity). Since N is typically 100,000 or more, this is prohibitively expensive. Thus, we compute sets of keys based on lexical similarity and compare only entities that are likely to match.

Key characteristics of this cross-document entity disambiguation algorithm, especially relative to other such methods (Bagga and Baldwin 1998; Gooi and Allan 2004; Huang et al. 2003; Kalashnikov and Mehrotra 2005; Mann and Yarowsky 2003; Mihalcea 2003; Ravin and Kazi 1999) are:

- Recognizes when identical names correspond to distinct entities.
- Recognizes when different names (including spelling and transliteration variations) correspond to a single entity.
- Uses many different sources of context as evidence.
- High disambiguation performance.
- High computational throughput.

3 Performance

We have tested our entity disambiguation system on several semi-structured and unstructured text data sets. Here, we report the performance on a set of 48,566 recent Pakistan News Service (<http://paknews.com/>) documents. This English-language newspaper focuses on the Middle East and consequently includes numerous transliterated entity names. Furthermore, numerous articles were written by amateurs and person who speak English as a second language. Thus, there is substantial variation in capitalization, punctuation, grammar, and spelling – characteristics that make NER challenging.

The named entity recognition process identified a total of 900,000 location, organization, and person mentions as shown in Table 2.

Entity Type	Number of Mentions
LOCATION	393,799
ORGANIZATION	248,943
PERSON	252,423

Table 2. Number of mentions by entity type in the data set.

Within-document entity disambiguation reduced the number of items to deal with to approximately 530,000 as shown in Table 3. As mentioned earlier, the *accuracy* of these merge operations exceeds 99%. This level of performance is possible because the task is not difficult (much easier than pronoun co-reference resolution, for example) and because some potential merge operations were not carried out at this stage since they can be handled more accurately (due to

more extensive statistics available) in the cross-document disambiguation process.

Entity Type	Document-level Entities
LOCATION	209,872
ORGANIZATION	180,095
PERSON	145,064

Table 3. Number of document-level entities (after within-document entity disambiguation) by entity type.

Cross-document entity disambiguation reduced the number of distinct persons to 55,477 with 36,204 distinct standardized names. Cross-document entity disambiguation is not currently implemented for locations and organizations, so further results are not available for these entity types.

Thus, 89,587 merge operations were performed in this step. Inspection of a representative sample of the entities shows that almost all of the merge operations (>95%) were correct. The system performs better when there is a lot of information than when confronted by a paucity of information. For example, the entity `p_pervez_musharraf_001_p` occurs in 8,187 documents. That is a lot of context for one entity! Furthermore, “Pervez” and “Musharraf” are fairly distinctive and uncommon names. All 8,186 merge operations for this entity seem to be correct. The system found over 40 spelling variations of this name, some of which are listed in Table 4.

Gneral Pervez Musharraf	Pervaiz Musharraf
Musharraf	Pervez Musharraf Firday
Parvaiz Musharraf	Pervez Musharrafs
Parvez Musharraf	Pervez Mushharraf
Perevz Musharraf	Prvez Musharraf

Table 4. Some variations on the name “Pervez Musharraf” found in the data set.

Similarly, thorough manual inspection reveals that the system correctly assigned each of 245 document-level mentions of variants of “Yasser Arafat” (listed in Table 5) to the correct of two entities mentioned in the introduction, the cricket player and the Palestinian leader.

p_yasir_arafat_001_p	p_yasser_arafat_001_p
Arafat	Arafat
PNS Yasir Arafat	Yasar Arafat
Yasar Arafat	Yasir Arafat
Yasir Arafat	Yasser Arafat
Yasir Arfat	Yasser Arfat

Table 5. Variations on the name “Yasser Arafat” found in the data set, and unique entity to which mentions were assigned.

Disambiguation of entities that occur in few documents is more difficult, especially for common names such as “Abdul Khan”. In fact, the system found 81 distinct persons (occurring in 153 documents) named “Abdul Khan”.

Several techniques were used to arrive at the above accuracy figures. The cross-document entity disambiguation

software accepts ground truth information and logs all merge operations involving the labeled mentions, enabling automated accuracy and coverage measurement. We separately checked the accuracy on frequent, moderately frequent, and rare entities (as determined by the system), and combined the results for the overall (>95%) estimate. Name statistics sometimes obviated the need for exhaustively examining documents. For example, we found only four persons named “Musharraf” in the data set: Begum Sehba Musharraf, Javed Musharraf, Pervez Musharraf, and Zarin Musharraf-ud-Din. We *assume* that all mentions (including obvious spelling variations) of “Pervez Musharraf” refer to the same person. The graphical user interface described below was extremely helpful in this exploration.

The number of potential merge operations that the system *did not* carry out for pairs of entities with identical standard names is 19,273. Some (most) of these operations were correctly *not* carried out. Furthermore, some entities with differing standard names should have been merged. Nonetheless, this approximates an upper limit on the number of missed merge operations. Referring back to the aforementioned 89,587 merge operations that *were* performed, it appears that the system carried out well in excess of 80% (perhaps more like 90%) of the merge operations of two document-level entities that correspond to a single unique person in the real world.

We are aware of only one reference data set for benchmarking cross-document entity disambiguation results, the *John Smith corpus* (Bagga and Baldwin 1998), but were not able to obtain access to it in time to include results here. The John Smith corpus allows evaluation of only a subset of the cross-document entity disambiguation functionality. Documents with typing errors, transliteration variations, and mentioning “Smith” without a given name are not included. A more extensive reference corpus for cross-document entity disambiguation would help assess and advance the state of the art.

The current prototype disambiguation system’s Perl implementation is not optimized for speed. Nonetheless, single-threaded cross-document disambiguation for the entire PakNews corpus takes less than 30 minutes on a moderate Linux workstation, less than the time consumed by our NER. Implementation in C typically results in a factor of ten speed-up. Combined with additional hardware, it is clearly possible to carry out cross-document entity disambiguation in reasonable time on very large corpora.

A demonstration system with a graphical user interface for browsing disambiguation of entities and their relations is available by request from the author. As shown in Figure 1, the system enables selection of documents that mention a particular entity (as opposed to name), clustered display of entities, and more.

The data structures used both in the disambiguation as well as in the demonstration system facilitate incremental refinement (manual merging of entities or separation of entities that should not have been merged) by the user.

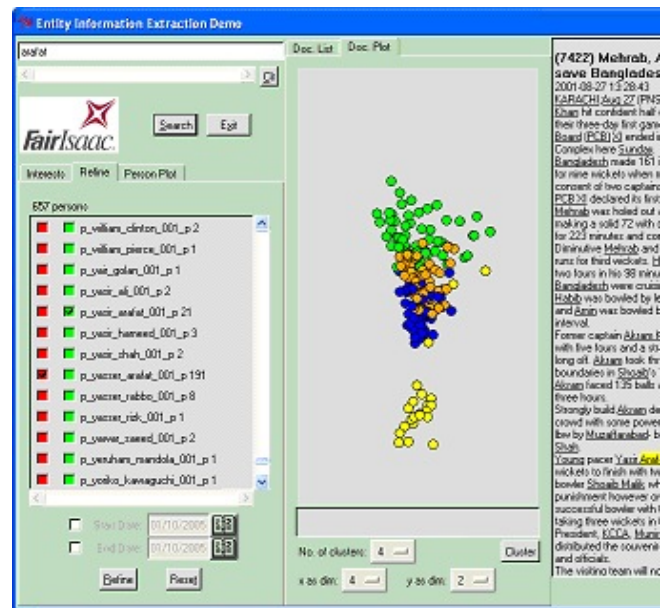


Figure 1. Demonstration system user interface.

4 Benefits to NER and Subsequent Processing

4.1 Named Entity Recognition

Most NER systems utilize only locally-available information for determining entity boundaries and entity types. Information from outside the current sentence or document is ignored. Entity disambiguation makes it possible to utilize this information to refine the NER results.

For example, if the entity disambiguation system determines that two mentions, “Dean Smith” and “Michael Smith” (whose title is “Dean”), correspond to the same entity, it is possible to correct the NER output to recognize the first “Dean” as a title rather than a given name.

Our system explicitly carries out this process at the stage of within-document entity disambiguation and in some cases (e.g. Table 4) also implicitly achieves this effect via cross-document entity disambiguation.

4.2 Relation Extraction

It is possible to find relation strings in text and process them without understanding the underlying entities. For example, the phrase “Chris Smith and his son David” may be parsed as indicating a family relationship between a person named “Chris Smith” and a person named “David”. However, it is difficult to obtain a confidence level for this relation (“David” might instead be the son of some other person in the example) based on a single relation mention.

Cross-document entity disambiguation makes it possible to define a new “strength of relation” assertion. A pair of entities that occurs together in many documents is related, period. This is one of several relation types that is exposed via our demonstration system (Figure 1). Similarly, if multiple documents mention the same relation between two entities, our confidence in this relation increases. This provides a mechanism for defining a relation confidence score.

4.3 Link Analysis

If “Chris” has a son named “David” and “David” has a son named “Evan”, does that mean that “Chris” and “Evan” are related? Only if the two “Davids” are the same person!

Entity disambiguation trivially turns a set of relation assertions derived from plain text into a linkage graph. Without entity disambiguation, it is impossible to generate this linkage graph. Entity disambiguation is a necessary technology for enabling fully automated link analysis from plain text data.

4.4 Inference

One of our partners was recently surprised to find that the “Chashma Nuclear Power Plant” “is located in” “Newark, NJ” (rather than in India). He was attempting to reason on assertions extracted from plain text without using entity disambiguation. “Chashma Nuclear Power Plant” “is a kind of” “plant” and “plant” “is located in” “Newark”. The problem, of course, is that the last “plant” refers to a specific plant that is not the same as the “Chashma Nuclear Power Plant”. Entity disambiguation would label the two plants as distinct entities and eliminate the problem.

In general, we do not reason with words – we reason with logic, and logic may be expressed in words. Similarly, inference does not operate on words – it operates on real-world concepts, object, events, and so on. Entity disambiguation provides the model that links text mentions to actual entities and thus makes it possible to reason over real-world objects based on information extracted from text.

5 Conclusions

Intelligence analysts are keenly interested in coalescing information about entities of interest across sizable collections of free-form text and in exploring relations between actual entities. This is very different from retrieving documents that contain a particular name (entity mention). Furthermore, the ability to determine which mentions refer to which real-world entities is absolutely essential for automatic link analysis and inference over information extracted from text.

This paper describes an implementation of this capability. The disambiguation performance is high in the many-to-many scenario that includes spelling variations and multiple persons with the same name. An efficient search technique provides adequate throughput for applying the methodology to large real-world data sets. A demonstration system is available, and the technology is offered for integration in intelligence analysis products.

Acknowledgment

This material is based on work funded in whole or in part by the U.S. Government (ARDA NIMD). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Government.

References

- Bagga, A. and Baldwin, B. 1998. Entity-based Cross-document Coreferencing Using the Vector Space Model. 17th International Conference on Computational Linguistics (CoLing-ACL). Montreal, Canada. 10-14 August, 1998, 79-85.
- Bikel, D. M.; Miller, S.; Schwartz, R. and Weischedel, R. 1997. Nymble: a High-Performance Learning Name-finder. Fifth Conference on Applied Natural Language Processing. Washington, D.C. 31 March - 3 April, 1997, 194-201.
- Caid, W. and Oing, P. 1997. System and Method of Context Vector Generation and Retrieval. U.S. Patent No. 5,619,709.
- Gooi, C. H. and Allan, J. 2004. Cross-Document Coreference on a Large Scale Corpus. Human Language Technology Conference (HLT-NAACL). Boston, Massachusetts. 2-7 May, 2004, 9-16.
- Hobbs, J. 1978. Resolving Pronoun References. *Lingua* 44(4): 311-338.
- Huang, F.; Vogel, S. and Waibel, A. 2003. Automatic Extraction of Named Entity Translingual Equivalence Based on Multi-feature Cost Minimization. ACL-03 Workshop on Multilingual and Mixed-language Named Entity Recognition. Sapporo, Japan. 12 July, 2003, 9-16.
- Kalashnikov, D. V. and Mehrotra, S. 2005. A Probabilistic Model for Entity Disambiguation Using Relationships. SIAM International Conference on Data Mining (SDM). Newport Beach, California. 21-23 April, 2005.
- Knuth, D. E. 1998. *The Art of Computer Programming, Volume 3: Sorting and Searching*. Addison-Wesley Professional.
- Mann, G. S. and Yarowsky, D. 2003. Unsupervised Personal Name Disambiguation. Conference on Computational Natural Language Learning (CoNLL). Edmonton, Canada. 31 May - 1 June, 2003, 33-40.
- Mihalcea, R. 2003. The Role of Non-Ambiguous Words in Natural Language Disambiguation. Conference on Recent Advances in Natural Language Processing (RANLP). Borovetz, Bulgaria. 10-12 September, 2003, .
- Mitchell, A.; Strassel, S.; Przybocki, P.; Davis, J. K.; Doddington, G.; Grishman, R.; Meyers, A.; Brunstein, A.; Ferro, L. and Sundheim, B. 2004. *Annotation Guidelines for Entity Detection and Tracking (EDT), Version 4.2.6*. <http://www ldc.upenn.edu/Projects/ACE/>.
- Ravin, Y. and Kazi, Z. 1999. Is Hillary Rodham Clinton the President? Disambiguating Names across Documents. ACL 1999 Workshop on Coreference and Its Applications. College Park, Maryland. 22 June, 1999, 9-16.