

Names: A New Frontier in Text Mining

Frankie Patman¹ and Paul Thompson²

¹Language Analysis Systems, Inc.
2214 Rock Hill Rd., Herndon, VA 20170
frankie@las-inc.com

²Institute for Security Technology Studies
Dartmouth College, Hanover, NH 03755
Paul.Thompson@dartmouth.edu

Abstract. Over the past 15 years the government has funded research in information extraction, with the goal of developing the technology to extract entities, events, and their interrelationships from free text for further analysis. A crucial component of linking entities across documents is the ability to recognize when different name strings are potential references to the same entity. Given the extraordinary range of variation international names can take when rendered in the Roman alphabet, this is a daunting task. This paper surveys existing technologies for name matching and for accomplishing pieces of the cross-document extraction and linking task. It proposes a direction for future work in which existing entity extraction, coreference, and database name matching technologies would be harnessed for cross-document coreference and linking capabilities. The extension of name variant matching to free text will add important text mining functionality for intelligence and security informatics toolkits.

1 Introduction

Database name matching technology has long been used in criminal investigations [1], counter-terrorism efforts [2], and in a wide variety of government processes, e.g., the processing of applications for visas. With this technology a name is compared to names contained in one or more databases to determine whether there is a match. Sometimes this matching operation may be a straightforward exact match, but often the process is more complicated. Two names may not match exactly for a wide variety of reasons and yet still refer to the same individual [3]. Often a name in a database comes from one field of a more complete database record. The values in other fields, e.g., social security number, or address, can be used to help match names which are not exact matches. The context from the complete record helps the matching process.

In this paper we propose the design of a system that would extend database name matching technology to the unstructured realm of free text. Over the past 15 or so years the federal government has funded research in information extraction, e.g., the Message Understanding Conferences [4], Tipster [5], and Automatic Content

Extraction [6]. The goal of this research has been to develop the technology to extract entities, events, and their interrelationships, from free text so that the extracted entities and relationships can be stored in a relational database, or knowledgebase, to be more readily analyzed. One subtask during the last few years of the Message Understanding Conference was the Named Entity Task in which personal and company names, as well as other formatted information, was extracted from free text. The system proposed in this paper would extract personal and company names from free text for inclusion in a database, an information extraction template, or automatically marked up XML text [7]. It would expand link analysis capabilities by taking into account a broad and more realistic view of the types of name variation found in texts from diverse sources. The sophisticated name matching algorithms currently available for matching names in databases are equally suited to matching name strings drawn from text. Analogous to the way in which the context of a full database record can assist in the name matching process, in the free text application, the context of the full text of the document can be used not only to help identify and extract names, but also to match names, both within a single document and across multiple documents.

2 Database Name Matching

Name matching can be defined as the process of determining whether two name strings are instances of the same name. It is a component of entity matching but is distinct from that larger task, which in many cases requires more information than a name alone. Name matching serves to create a set of candidate names for further consideration—those that are variants of the query name. ‘Al Jones’, for example, is a legitimate variant of ‘Alfred Jones,’ ‘Alan Jones,’ and ‘Albert Jones.’ Different processes from those involved in name matching will often be required to equate entities, perhaps relation to a particular place, organization, event, or numeric identifier. However, without a sufficient representation of a name (the set of variants of the name likely to occur in the data), different mentions of the same entity may not be recognized.

Matching names in databases has been a persistent and well-known problem for years [8]. In the context of the English-speaking world alone, where the predominant model for names is a given name, an optional middle name, and a surname of Anglo-Saxon or Western European origin, a name can have any number of variant forms, and any or all of these forms may turn up in database entries. For example, Alfred James Martin can also be A. J. Martin; Mary Douglas McConnell may also be Mary Douglas or Mary McConnell or Mary Douglas-McConnell; Jack Crowley and John Crowley may both refer to the same person; the surnames Laury and Lowrie can have the same pronunciation and may be confused when names are taken orally; jSmith is a common typographical error entered for the name Smith. These familiar types of name variation pose non-trivial difficulties for automatic name matching, and numerous systems have been devised to deal with them (see [3]).

The challenges to name matching are greatly increased when databases contain names from outside the Anglo-American context. Consider some common issues that arise with names from around the world.

- In China or Korea, the surname comes first, before the given name. Some people may maintain this format in Western contexts, others may reverse the name order to fit the Western model, and still others may use either. The problem is compounded further if a Western given name is added, since there is no one place in the string of names where the additional name is required to appear.
Ex: Yi Kyung Hee ~ Kyung Hee Yi ~ Kathy Yi Kyung Hee ~ Yi Kathy Kyung Hee ~ Kathy Kyung Hee Yi
- In some Asian countries, such as Indonesia, many people have only one name; what appears to be a surname is actually the name of the father. Names are normally indexed by the given name. Ex: former Indonesian president Abdurrahman Wahid is Mr. Abdurrahman (Wahid being the name of his father).
- A name from some places in the Arab world may have many components showing the bearer's lineage, and none of these is a family name. Any one of the name elements other than the given name can be dropped. Ex: Aziz Hamid Salim Sabah ~ Aziz Hamid ~ Aziz Sabah ~ Aziz
- Hispanic names commonly have two surnames, but it is the first of these rather than the last that is the family name. The final surname (which is the mother's family name) may or may not be used. Ex: Jose Felipe Ortega Ballesteros ~ Jose Felipe Ortega, but is less likely to refer to the same person as Jose Felipe Ballesteros
- There may be multiple standard systems for transliterating a name from a native script (e.g. Arabic, Chinese, Hangul, Cyrillic) into the Roman alphabet, individuals may make up their own Roman spelling on the fly, or database entry operators may spell an unfamiliar name according to their own understanding of how it sounds. Ex: Yi ~ Lee ~ I ~ Lie ~ Ee ~ Rhee
- Names may contain various kinds of affixes, which may be conjoined to the rest of the name, separated from it by white space or hyphens, or dropped altogether. Ex: Abdalsharif ~ Abd al-Sharif ~ Abd-Al-Sharif ~ Abdal Sharif; al-Qaddafi ~ Qaddafi

Systems for overcoming name variation search problems typically incorporate one or more of (1) a non-culture-specific phonetic algorithm (like Soundex¹ or one of its refinements, e.g. [9]); (2) allowances for transposed, additional, or missing characters; (3) allowances for transposed, additional or missing name elements and for initials and abbreviations; and (4) nickname recognition. See [10] for a recent example. Less commonly, culture-specific phonetic rules may be used.

The most serious problem for name-matching software is the wide variety of naming conventions represented in modern databases, which reflects the multi-cultural composition of many societies. Name-matching algorithms tend to take a one-size-fits-all approach, either by underestimating the effects of cultural variation,

¹ Soundex, the most well-known algorithm for variant name searching in databases, is a phonetics-based system patented in 1918. It was devised for use in indexing the 1910 U.S. census data. The system groups consonants into sets of similar sounds (based on American names reported at the time) and assigns a common code to all names beginning with the same letter and sharing the same sequence of consonant groups. Soundex does not accommodate certain errors very well, and groups many highly dissimilar names under the same code. See [11].

or by assuming that names in any particular data source will be homogenous. This may give reasonable results for names that fit one model, but may perform very poorly with names that follow different conventions. In the area of spelling variation alone, which letters are considered variants of which others differs from one culture to the next. In transcribed Arabic names, for example, the letters “K” and “Q” can be used interchangeably; “Qadafi” and “Kadafi” are variants of the same name. This is not the case in Chinese transcriptions, however, where “Kuan” and “Quan” are most likely to be entirely different names. What constitutes similarity between two name strings depends on the culture of origin of the names, and typically this must be determined on a case-by-case basis rather than across an entire data set.

Language Analysis Systems, Inc. (LAS) has implemented a number of approaches to coping with the wide array of multi-cultural name forms found in databases. Names are first submitted to an automatic analysis process, which determines the most likely cultural/linguistic origin of the name (or, at the discretion of the user, the culture of origin can be manually chosen). Based on this determination, an appropriate algorithm or set of rules is applied to the matching process. LAS technologies include culturally sensitive search systems and processes for generating variants of names, among others. Some of the LAS technologies are briefly discussed below.

Automatic Name Analysis: The name analysis system (NameClassifier™) contains a knowledge base of information about name strings from various cultures. An input name is compared to what is known about name strings from each of the included cultures, and the probability of the name’s being derived from each of the cultures is computed. The culture with the highest score is assigned to the input name. The culture assignment is then used by other technologies to determine the most appropriate name-matching strategy.

NameVariantGenerator™: Name variant generation produces orthographic and syntactic variants of an input string. The string is first assigned a culture of origin through automatic name analysis. Culture-specific rules are then applied to the string to produce a regular expression. The regular expression is compared to a knowledge base of frequency information about names drawn from a database of over 750,000,000 names. Variant strings with a high enough frequency score are returned in frequency-ranked order. This process creates a set of likely variants of a name, which can then be used for further querying and matching.

NameHunter™: NameHunter™ is a search engine that computes the similarity of two name strings based on orthography, word order, and number of elements in the string. The thresholds and parameters for comparison differ depending on the culture assignment of the input string. If a string from the database has a score that exceeds the thresholds for the input name culture, the name is returned. Returns are ranked relative to each other, so that the highest scoring strings are presented first. NameHunter allows for noisy data; thresholds can be tweaked by the user to control the degree of noise in returns.

MetaMatch™: MetaMatch™ is a phonetic-based name retrieval system. Entry strings are first submitted to automatic name analysis for a culture assignment. Strings are then transformed to phonetic representations based on culture-specific rules, which are then stored in the database along with the original entry. Query strings are similarly processed, and the culture assignment is retained to determine the particular

parameters and thresholds for comparison. A similarity algorithm based on linguistic principles is used to determine the degree of similarity between query and entry strings [12]. Returns are presented in ranked order. This approach is particularly effective when name entries have been drawn from oral sources, such as telephone conversations.

NameGenderizer™: This module returns the most likely gender for a given name based on frequency of assignment of the name to males or females.

A major advantage of the technologies developed by LAS is that a measure of similarity between name forms is computed and used to return names in order of their degree of similarity to the query term. An example of the effectiveness of this approach over a Soundex search is provided in Fig.1 in the Appendix.

3 Named Entity Extraction

The task of named entity recognition and extraction is to identify strings in text that represent names of people, organizations, and places. Work in this area began in earnest in the mid-eighties, with the initiation of the Message Understanding Conferences (MUC). MUC is largely responsible for the definition of and specifications for the named entity extraction task as it is understood today [4].

Through MUC-6 in 1995, most systems performing named entity extraction were based on hand-built patterns that recognized various features and structures in the text. These were found to be highly successful, with precision and recall figures reaching 97% and 96%, respectively [4]. However, the systems were trained exclusively on English-language newspaper articles with a fixed set of domains, leaving open the question of how they would perform on other text sources. Bikel et al. [13] found that rules developed for one newswire source had to be adapted for application to a different newswire service, and that English-language rules were of little use as a starting point for developing rules for an unrelated language like Chinese. These systems are labor-intensive and require people trained in text analysis and pattern writing to develop and maintain rule sets.

Much recent work in named entity extraction has focused on statistical/probabilistic approaches (e.g., [14], [15], [13], [16]). Results in some cases have been very good, with F-measure scores exceeding 94%, even for systems gathering information from the least computationally expensive sources, such as punctuation, dictionary look-up, and part-of-speech taggers [15]. Borthwick et al. [14] found that by training their system on outputs tagged by hand-built systems (such as SRA's NameTag extractor), scores improved to better than 97%, exceeding the F-measure scores of hand-built systems alone, and rivaling scores of human annotators. These results are very promising and suggest that named entity extraction can be usefully applied to larger tasks such as relation detection and link analysis (see, for example, [17]).

4 Intra- and Inter-document Coreference

The task of determining coreference can be defined as “the process of determining whether two expressions in natural language refer to the same entity in the world,” [18]. Expressions handled by coreference systems are typically limited to noun phrases of various types—including proper names—and pronouns. This paper will consider only coreference between proper names.

For a human reader, coreference processes take place within a single document as well as across multiple documents when more than one text is read. Most coreference systems deal only with coreference within a document (see [19], [20], [21], [18], [22]). Recently, researchers have also begun work on the more difficult task of cross-document coreference ([23], [24], [25]).

Bagga [26] offers a classification scheme for evaluating coreference types and systems for performing coreference resolution, based in part on the amount of processing required. Establishing coreference between proper names was determined to require named entity recognition and generation of syntactic variants of names. Indeed, the coreference systems surveyed for this paper treat proper name variation (apart from synonyms, acronyms, and abbreviations) largely as a syntactic problem. Bontcheva et al., for example, allow name variants to be an exact match, a word token match that ignores punctuation and word order (e.g., “John Smith” and “Smith, John”), a first token match for cases like “Peter Smith” and “Peter,” a last token match for e.g., “John Smith” and “Smith,” a possessive form like “John’s,” or a substring in which all word tokens in the shorter name are included in the longer one (e.g., “John J. Smith” and “John Smith”).

Depending on the text source, name variants within a single document are likely to be consistent and limited to syntactic variants, shortened forms, and synonyms, such as nicknames.² One would expect intra-document coreference results for proper names under these circumstances to be fairly good. Bontcheva et al. [19] obtained precision and recall figures ranging from 94%-98% and 92%-95%, respectively, for proper name coreferences in texts drawn from broadcast news, newswire, and newspaper sources.³

Bagga and Baldwin [23] also report very good results (F-measures up to 84.6%) for tests of their cross-document coreference system, which compares summaries created for extracted coreference chains. Note, however, that their reported research looked only for references to entities named “John Smith,” and that the focus of the cross-document coreference task was maintaining distinctions between different entities with the same name. Research was conducted exclusively on texts from the New York Times. Nevertheless, their work demonstrates that context can be effectively used for disambiguation across documents. Ravin and Kazi [24] focus on both distinguishing different entities with the same name and merging variant names

² Note, however, that even within a document inconsistencies are not uncommon, especially when dealing with names of non-European origin. A Wall Street Journal article appearing in January 2003 referred to Mohammed Mansour Jabarah as Mr. Jabarah, while Khalid Sheikh Mohammed was called Mr. Khalid.

³ When items other than proper names are considered for coreference, scores are much lower than those reported by Bontcheva et al. for proper names. The highest F-measure score for coreference at the MUC-7 competition was 61.8%. This figure includes coreference between proper names, various types of noun phrases, and pronouns.

referring to a single entity. They use the IBM Context Thesaurus to compare the contexts in which similar names from different documents are found. If there is enough overlap in the contextual information, the names are assumed to refer to the same entity. Their work was also limited to articles from the New York Times and the Wall Street Journal, both of which are edited publications with a high degree of internal consistency.

Across documents from a wide variety of sources, consistent name variants cannot be counted on, especially for names originating outside the Anglo/Western European tradition. In fact, the many types of name variation commonly found in databases can be expected. A recent web search on Google for texts about Muammar Qaddafi, for example, turned up thousands of relevant pages under the spellings Qathafi, Kaddafi, Qadafi, Gadafi, Gaddafi, Kathafi, Kadhafi, Qadhafi, Qazzafi, Kazafi, Qaddafy, Qadafy, Quadhaffi, Gadhhdafi, al-Qaddafi, Al-Qaddafi, and Al Qaddafi (and these are only a few of the variants of this name known to occur). A coreference system that can be of use to agencies dealing with international names must be able to recognize name strings with this degree of variation as potential instances of a single name.

Cross-document coreference systems currently suffer from the same weakness as most database name search systems. They assume a much higher degree of source homogeneity than can be expected in the world outside the laboratory, and their analysis of name variation is based on an Anglo/Western European model. For the coreference systems surveyed here, recall would be a considerable problem within a multi-source document collection containing non-Western names. However, with an expanded definition of name variation, constrained and supplemented by contextual information, these coreference technologies can serve as a starting point for linking and disambiguating entities across documents from widely varying sources.

5 Name Text Mining Support for Visualization, Link Analysis, and Deception Detection

Commercial and research products for visualization and link analysis have become widely available in recent years, e.g., Hyperbolic Tree, or Star Tree [27], SPIRE [28], COPLINK [29], and InfoGlide [30]. Visualization and link analysis continues to be an active area of on-going research [31]. Some current tools have been incorporated into systems supporting intelligence and security informatics. For example, COPLINK [29] makes use of several visualization and link analysis packages, including i2's [32] Analyst Notebook. Products such as COPLINK and InfoGlide also support name matching and deception detection. These tools make use of sophisticated statistical record linkage, e.g. [33], and have well developed interfaces to support analysts [32, 29]. Chen et al. [29] note that COPLINK Connect has the built-in capability for partial and phonetic-based name searches. It is not clear from the paper, however, what the scope of coverage is for phonetically spelled names, or how this is implemented.

Research software and commercial products have been developed, such as those presented in [34, 30], which include modules that detect fraud in database records. These applications' foci model ways that criminals, or terrorists, typically alter records to disguise their identity. The algorithms used by these systems could be

augmented by taking into account a deeper multi-cultural analysis of names, as discussed in section 2.

6 Procedure for a Name Extraction and Matching Text Mining Module

In this section a procedure is presented for name extraction and matching within and across documents. This algorithm could be incorporated in a module that would work with an environment such as COPLINK. The basic algorithm is as follows.

Within document:

1. Perform named entity extraction.
2. Establish coreference between name mentions within a single document, creating an equivalence class for each named entity.
3. Discover relations between equivalence classes within each document
4. Find the longest canonical name string in each equivalence class.
5. Perform automatic name analysis on canonical names using NameClassifier; retain culture assignment.
6. Generate variant forms of canonical names according to culture-specific criteria using NameVariantGenerator.

Across documents:

7. For each culture identified during name analysis, match sets of canonical name variants belonging to that culture against each other; for each pair of variant sets considered, if there are no incompatible (non-matching) members in the sets, mark as potential matches (e.g., Khalid bin (son of) Jamal and Khalid abu (father of) Jamal would be incompatible).
8. For potential name set matches, use a context thesaurus like that described in [24] to compare contexts where the names in the equivalence classes are found; if there are enough overlapping descriptions, merge the equivalence classes for the name sets (which will also expand the set of relations for the class to include those found in both documents); combine variant sets for the two canonical name strings into a single set, pruning redundancies.
9. For potential name set matches where overlapping contextual descriptions do not meet the minimum threshold, mark as a potential link, but do not merge.
10. Repeat process from #7 on for each pair of variant sets, until no further comparisons are possible.

This algorithm could be implemented within a software module of a larger text mining application. The simplest integration of this algorithm would be as a module that extracted personal names from free text and stored the extracted names and relationships in a database. As discussed by [7], it would also be possible to use this algorithm to annotate the free text, in addition to creating database entries. This automatic markup would provide an interface for an analyst which would show not only the entities and their relationships, but also preserve the context of the surrounding text.

7 Research Issues

This paper proposes an extension of linguistically-based, multi-cultural database name matching functionality to the extraction and matching of names from full text documents. To accomplish such an extension implies an effective integration of database and document retrieval technology. While this has been an on-going research topic in academic research [35, 36] and has received attention from major relational database vendors such as Oracle, Sybase, and IBM, effective integration has not yet been achieved, in particular in the area of intelligence and security informatics [37]. Achieving the sophistication of database record matching for names extracted from free text implies advances in text mining [38, 39, 40, 41].

One useful structure for supporting cross document name matching would be an authority file for named entities. Library catalogs maintain authority files which have a record for each author, showing variant names, pseudonyms, and so on. An authority file for named entity extraction could be built which would maintain a record for each entity. The record could start with information about the entity extracted from database records. When the named entity was found in free text, contextual information about the entity could be extracted and stored in the authority file with an appropriate degree of probability in the accuracy of the information included. For example, a name followed by a comma-delimited parenthetical expression, is a reasonably accurate source of contextual information about an entity, e.g., “X, president of Y, resigned yesterday”.

A further application of linguistic/cultural classification of names could be to tracking interactions between groups of people where there is a strong association between group membership and language. For example, an increasing number of police reports in which both Korean and Cambodian names are found in the same documents might indicate a pattern in Asian crime ring interactions.

Finally, automatic recognition of name gender could be used to support the process of pronominal coreference.

Work is underway to provide a quantitative comparison of key-based name matching systems (such as Soundex) with other approaches to name matching. One of the hindrances to effective name matching system comparisons is the lack of generally accepted standards for what constitutes similarity between names. Such standards are difficult to establish in part because the definition of similarity changes from one user community to the next. A standardized metric for the evaluation of degrees of correlation of name search results, and a means for using this metric to measure the usefulness of different name search technologies is sorely needed.

This paper has focused on personal name matching. Matching of other named entities, such as organizations, is also of interest for intelligence and security informatics. While different matching algorithms are needed, extending company name matching, or other entity matching, to free text will also be useful. One promising research direction integrating database, information extraction, and document retrieval that could support effective text mining of names is provided by work on XIRQL [7].

8 Conclusion

Effective tools exist for multi-cultural database name matching and this technology is becoming available in analytic tool kits supporting intelligence and security informatics. The proportion of data of interest to intelligence and security analysts that is contained in databases, however, is very small compared to the amount of data available in free text and audio formats. The extension of name extraction and matching to free text and audio will add important text mining functionality for intelligence and security informatics toolkits.

References

1. Taft, R.L.: Name Search Techniques. Special Rep. No. 1. Bureau of Systems Development, New York State Identification and Intelligence System, Albany (1970)
2. Verton, D.: Technology Aids Hunt for Terrorists. *Computer World*, 9 September (2002)
3. Borgman, C.L., Siegfried, S.L.: Getty's Synoname and Its Cousins: A Survey of Applications of Personal Name-Matching Algorithms. *Journal of the American Society for Information Science*, Vol. 43 No. 7. (1992) 459–476
4. Grishman, R., Sundheim, B.: Message Understanding Conference – 6: A Brief History. In: *Proceedings of the 16th International Conference on Computational Linguistics*. Copenhagen (1999)
5. DARPA. Tipster Text Program Phase III Proceedings. Morgan Kaufmann, San Francisco (1999)
6. National Institute of Standards and Technology. ACE-Automatic Content Extraction Information Technology Laboratories. <http://www.itl.nist.gov/iad/894.01/tests/ace/index.htm> (2000)
7. Fuhr, N.: XML Information Retrieval and Extraction [to appear]
8. Hermansen, J.C.: Automatic Name Searching in Large Databases of International Names. Georgetown University Dissertation, Washington, DC (1985)
9. Holmes, D., McCabe, M.C.: Improving Precision and Recall for Soundex Retrieval. In: *Proceedings of the 2002 IEEE International Conference on Information Technology – Coding and Computing*. Las Vegas (2002)
10. Navarro, G., Baeza-Yates, R., Azevedo Arcoverde, J.M.: Matchsimile: A Flexible Approximate Matching Tool for Searching Proper Names. *Journal of the American Society for Information Science and Technology*, Vol. 54 No. 1 (2003) 3–15
11. Patman, F., Shaefer, L.: Is Soundex Good Enough for You? On the Hidden Risks of Soundex-Based Name Searching. Language Analysis Systems, Inc., Herndon (2001)
12. Lutz, R., Greene, S.: Measuring Phonological Similarity: The Case of Personal Names. Language Analysis Systems, Inc., Herndon (2002)
13. Bikel, D.M., Schwartz, R., Weischedel, R.M.: An Algorithm that Learns What's in a Name. *Machine Learning*, Vol. 34 No. 1-3. (1999) 211–231
14. Borthwick, A., Sterling, J., Agichtein, E., Grishman, R.: NYU: Description of the MENE Named Entity System as Used in MUC-7. In: *Proceedings of the Seventh Message Understanding Conference*. Fairfax (1998)
15. Baluja, S., Mittal, V.O., Sukthankar, R.: Applying Machine Learning for High Performance Named-Entity Extraction. *Pacific Association for Computational Linguistics* (1999)
16. Collins, M.,: Ranking Algorithms for Named-Entity Extraction: Boosting and the Voted Perceptron. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia (2002) 489–496

17. Zelenko, D., Aone, C., Richardella, A.: Kernel Methods for Relation Detection Extraction. *Journal of Machine Learning Research* [to appear]
18. Soon, W.M., Ng, H.T., Lim, D.C.Y.: A Machine Learning Approach to Coreference Resolution of Noun Phrases. *Association for Computational Linguistics* (2001)
19. Bontcheva, K., Dimitrov, M., Maynard, D., Tablin, V., Cunningham, H.: Shallow Methods for Named Entity Coreference Resolution. *TALN* (2002)
20. Harttrumpf, S.: Coreference Resolution with Syntactico-Semantic Rules and Corpus Statistics. In: *Proceedings of CoNLL-2001*. Toulouse (2001) 137–144
21. Ng, V., Cardie, C.: Improving Machine Learning Approaches to Coreference Resolution. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia (2002) 104–111
22. McCarthy, J.F., Lehnert, W.G.: Using Decision Trees for Coreference Resolution. In: Mellish, C. (ed.): *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* (1995) 1050–1055
23. Bagga, A., Baldwin, B.: Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In: *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics* (1998) 79–85
24. Ravin, Y., Kazi, Z. Is Hillary Rodham Clinton the President? Disambiguating Names Across Documents. In: *Proceedings of the ACL'99 Workshop on Coreference and Its Applications* (1999)
25. Schiffman, B., Mani, I., Concepcion, K.J.: Producing Biographical Summaries: Combining Linguistic Knowledge with Corpus Statistics. In: *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics* (2001) 450–457
26. Bagga, A.: Evaluation of Coreferences and Coreference Resolution Systems. In: *Proceedings of the First International Conference on Language Resources and Evaluation* (1998) 563–566
27. Inxight. A Research Engine for the Pharmaceutical Industry. <http://www.inxight.com>
28. Hetzler, B., Harris, W.M., Havre, S., Whitney, P.: Visualizing the Full Spectrum of Document Relationships. In: *Structures and Relations in Knowledge Organization*. *Proceedings of the 5th International ISKO Conference*. ERGON Verlag, Wurzburg (1998) 168–175
29. Chen, H., Zeng, D., Atabakhsh, H., Wyzga, W., Schroeder, J.: COPLINK: Managing law enforcement data and knowledge. *Communications of the ACM*, Vol. 46 No. 1 (2003)
30. InfoGlide Software. Similarity Search Engine: The Power of Similarity Searching. [http://www.infoglide.com/content/images/whitepapers.pdf\(2002\)](http://www.infoglide.com/content/images/whitepapers.pdf(2002))
31. American Association for Artificial Intelligence Fall Symposium on Artificial Intelligence and Link Analysis (1998)
32. i2. Analyst's Notebook. http://www.i2.co.uk/Products/Analysts_Notebook (2002)
33. Winkler, W.E.: The State of Record Linkage and Current Research Problems. Technical Report RR99/04. U.S. Census Bureau, <http://www.census.gov/srd/papers/pdf/rr99-04.pdf>
34. Wang, G., Chen, H., Atabakhsh, H.: Automatically Detecting Deceptive Criminal Identities [to appear]
35. Fuhr, N.: Probabilistic Datalog – A Logic for Powerful Retrieval Methods. In: *Proceedings of SIGIR-95*, 18th ACM International Conference on Research and Development in Information Retrieval (1995) 282–290
36. Fuhr, N.: Models for Integrated Information Retrieval and Database Systems. *IEEE Data Engineering Bulletin*, Vol. 19 No. 1. (1996)
37. Hooegeveen, M., van der Meer, K.: Integration of Information Retrieval and Database Management in Support of Multimedia Police Work. *Journal of Information Science*, Vol. 20 No. 2 (1994)
38. Institute for Mathematics and Its Applications. IMA Hot Topics Workshop: Text Mining. <http://www.ima.umn.edu/reactive/spring/tm.html> (2000)

- 39. KDD-2000 Workshop on Text Mining. The Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Boston (2000) <http://www-2.cs.cmu.edu/~dunja/WshKDD2000.html>
- 40. SIAM Text Mining Workshop. <http://www.cs.utk.edu/tmw02> (2002)
- 41. Text-ML 2002 Workshop on Text Learning. The Nineteenth International Conference on Machine Learning ICML-2002. Sydney (2002)

Appendix: Comparison of LAS MetaMatch™ Search Engine Returns with SQL-Soundex Returns

MetaMatch Results (10)			SQL-Soundex Results (138)	
Surname	ID	Similarity Score	Surname	ID
SADIQ	34820	1.000000 (E)	STACY	1619
SADIK	68371	0.968760 (E)	SHEETS	1677
SADAQ	42264	0.942580 (E)	SEITZ	2837
SIDDIQ	67875	0.930470 (E)	STUCKEY	3201
SIDDIQUI	14481	0.857021 (E)	STACEY	3429
SIDDIQUE	45416	0.857021 (E)	STACK	3474
SUDAK	49162	0.855240 (E)	STOCK	3493
SOTAK	67672	0.839400 (E)	STAGGS	3718
SODEK	60144	0.839400 (E)	SIDES	4411
SUTIC	77474	0.837420 (E)	STOCKS	5252
			SITES	5646
			SADOWSKI	6154
			SHATTUCK	6467
			SOUTHWICK	8692
			STAGG	8880

Fig. 1. These searches were conducted in databases containing common surnames found in the 1990 U.S. Census data. The surnames in the databases are identical. The MetaMatch database differs only in that the phonetic form of each surname is also stored. The exact match “Sadiq” was 54th in the list of Soundex returns. “Siddiqui” was returned by Soundex in 26th place. “Sadik” was 109th.