Improving Personal Name Search in the TIGR System

Keith J. Miller, Sarah McLeod, Elizabeth Schroeder, Mark Arehart, Kenneth Samuel, James Finley, Vanesa Jurica, John Polk

The MITRE Corporation
7515 Colshire Drive
McLean, VA 22102
USA
{keith, smcleod, eschroeder, marehart, samuel, jtfinley, vjurica, jpolk}@mitre.org

Abstract

This paper describes the development and evaluation of enhancements to the specialized information retrieval capabilities of a multimodal reporting system. The system enables collection and dissemination of information through a distributed data architecture by allowing users to input free text documents, which are indexed for subsequent search and retrieval by other users. This unstructured data entry method is essential for users of this system, but it requires an intelligent support system for processing queries against the data. The system, known as TIGR (Tactical Ground Reporting), allows keyword searching and geospatial filtering of results, but lacked the ability to efficiently index and search person names and perform approximate name matching. To improve TIGR's ability to provide accurate, comprehensive results for queries on person names we iteratively updated existing entity extraction and name matching technologies to better align with the TIGR use case. We evaluated each version of the entity extraction and name matching components to find the optimal configuration for the TIGR context, and combined those pieces into a named entity extraction, indexing, and search module that integrates with the current TIGR system. By comparing system-level evaluations of the original and updated TIGR search processes, we show that our enhancements to personal name search significantly improved the performance of the overall information retrieval capabilities of the TIGR system[1].

1 Introduction

Variation in person names presents a challenge to search technology in any language, and the problems are multiplied when the names are represented in non-native languages and scripts. For example, when names from Arabic are rendered in English, sources of variation include: sounds that occur in Arabic but not in English, dialect variation, ambiguity due to lack of vowels in most written Arabic, optional titles and particles, inconsistent segmentation into name parts, inconsistent application of transliteration standards, and typographical errors. The complexity of Arabic name structure is also a significant contributor to name variation. Arabic names are structurally more complex and generally contain more name parts than the typical Anglo name, with name parts providing different information value. Finally, it is not uncommon for an individual to use a subset of their full name in a particular context.

The difficulties in name search mentioned above are further compounded when the data to be searched is created by users who are generally not familiar with the naming system of the source language. Such is the case for the Tactical Ground Reporting (TIGR) system. This system is a multimodal geospatial reporting system that enables the collection and dissemination of information through a distributed data architecture (DARPA, 2009). TIGR documents are free text, entered in an unrestricted format by multiple users who are generally untrained in the transcription of foreign names. The documents also lack metadata indicating which persons are mentioned in the text.

The original TIGR system (Original TIGR) indexes these documents by token, and stores them for later retrieval. Users can search the saved documents using a very basic search algorithm. Original TIGR employs an exact matching algorithm that allows the Boolean operators AND, OR, and – (not). Queries containing multiple words but no Boolean operator only return documents with all words in the query. Because the system only handles exact matching, and because authors do not consistently spell foreign names within and across documents, it is possible that users are missing relevant information when searching the saved information. This paper describes how we used existing name matching and entity extraction technologies to enhance the name search capability in the TIGR system, as well as our methodology for carrying out component-level evaluations to ensure optimal performance of the entity extraction and name matching components,

[1] The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

and system-level evaluation to measure overall information retrieval improvement for the use case in question.

2 Development

To improve search results for Arabic names we used a two-part strategy to develop the initial prototype for rapid deployment.

First, we customized an entity extraction software package (Wellner, 2008) to identify person names in TIGR documents. This entity extraction package is based on a supervised machine learning approach using conditional random fields to classify data (Lafferty, 2001). Using this approach, we built an extraction model based on a training data set that we created by annotating existing TIGR documents. Because the performance of the entity extractor depends crucially on the quality of the training data, we created a set of "annotation guidelines" to increase consistency among annotators. These guidelines were created after an initial review of the TIGR texts, and took into account the TIGR mission objectives. Using the guidelines, we manually annotated a corpus of TIGR documents to denote person names within the text. We incrementally built models as the size of the annotated corpus grew, using 10-fold cross-validation to test the performance of these extraction models along the way. The results at each increment are reported in the Evaluation section. The integration of this extraction model into TIGR consists of running each new document through the entity extraction model as it is entered into the system, with the resulting person names being entered into a specialized person name index.

Second, we optimized an existing fuzzy name matching system specialized for Romanized Arabic names to meet performance standards required for integration into the TIGR system. The name matcher identifies names in the name-only index that match a name in a user query according to a predefined threshold, returning the documents containing the indexed names to the user. Since the TIGR system needs to return results in real time, further experiments were performed to increase the speed of name matching while maintaining its accuracy (Arehart, 2010).

Both the entity extraction model and the optimized name matching system were incorporated into TIGR. To expedite integration, the system was updated to use both the original inverted index and the new name indexing function side by side, rather than incorporating the name extraction capability into the previously-existing indexing function. Thus, the updated TIGR system uses both the existing search algorithm and fuzzy name matching algorithm to identify documents relevant to user queries. As discussed in the Future Work section, we plan to explore tighter integration of the entity extraction and name matching capabilities with the Original TIGR search in future versions.

3 Evaluation

Throughout the development process we evaluated the entity extraction and name matching components. Comparing evaluation metrics from one update to the next enabled us to determine which changes were effective in improving performance. Using the most effective versions of these components, we created the initial prototype for integration into the original TIGR system. Using a system-level, or end-to-end, evaluation process allowed us to compare the updated search prototype to the original TIGR configuration and to measure the level of improvement in name search. The following sections describe each evaluation step in more detail.

3.1 Data

As previously mentioned, the TIGR data consists of unstructured documents written by different authors. These documents range in length from one line to multiple pages, and vary widely in writing style and format. All documents have the same header format, though the amount of information filled out for each document varies. Though documents can contain multiple types of names, we focused on two types, which we will refer to as A and B. Type A names are person names that resemble email addresses and other forms of address, while type B names are person names as they normally appear in free text.

The annotation guidelines described in the Development section were created based on documents from the original TIGR system Following these guidelines, we annotated approximately 3,000 documents with metadata indicating which token or string of tokens is a person name, as well as whether the name was type A or B. This was done in case it was later decided to handle names of type A and B differently in the indexing process. For example, one option under consideration was to eliminate names of type A from the entity index. Additionally, we annotated a small set of 35 news wire documents with which we could create a baseline extraction model. This model was both out-of-domain and based on a small amount of training data, and so provided an absolute lower bound for entity extraction performance. We divided the annotated TIGR documents into two sets. The first set was used to train and test the entity extraction model, while the remainder of the annotated data was used for the end-to-end system evaluation..

3.2 Component-level Evaluation

3.2.1. Entity Extraction

To evaluate the entity extraction model, we used the Message Understanding Conference (MUC) methodology (Douthat, 1998), and the latest version of the MUC Scoring Software. This software package compares the test data to the corresponding ground truth data and counts the number of times the extraction model correctly and incorrectly tagged entities, with counts broken down by type of entity. Using these counts, the MUC software calculates precision and recall for each type of entity extracted as well as for all entities in the entire set of test documents, and also gives the F-score for the entire set.

We performed 10-fold cross validation to evaluate the performance of entity extraction models built at three points during the annotation process to observe how additional training data affected extraction performance. These different points are referred to as Iteration 1, Iteration 2, and Iteration 3. Iteration 1 used 778 documents, with 700 documents per training fold and 78 per testing fold. Iteration 2 used 1439 documents, with 1295 training documents per fold and 144 testing documents per fold. Iteration 3 used 1890 documents, with 1701 training documents and 189 testing documents per fold. The final set of 1890 training documents contained an average of 4.51 names per document. We ran the baseline model against each fold of the testing data for each iteration, and averaged those precision, recall, and F-scores across the 10 folds. We then averaged the 10 precision, recall, and F-score metrics for the entire test set at each iteration. Those scores are reported in Figure 1. As expected, the third iteration with the most training data had higher scores than the first two iterations. As expected, the baseline model performed very poorly, which can be attributed both to the small size of the baseline training corpus and the difference between news wire and TIGR document structure and content.

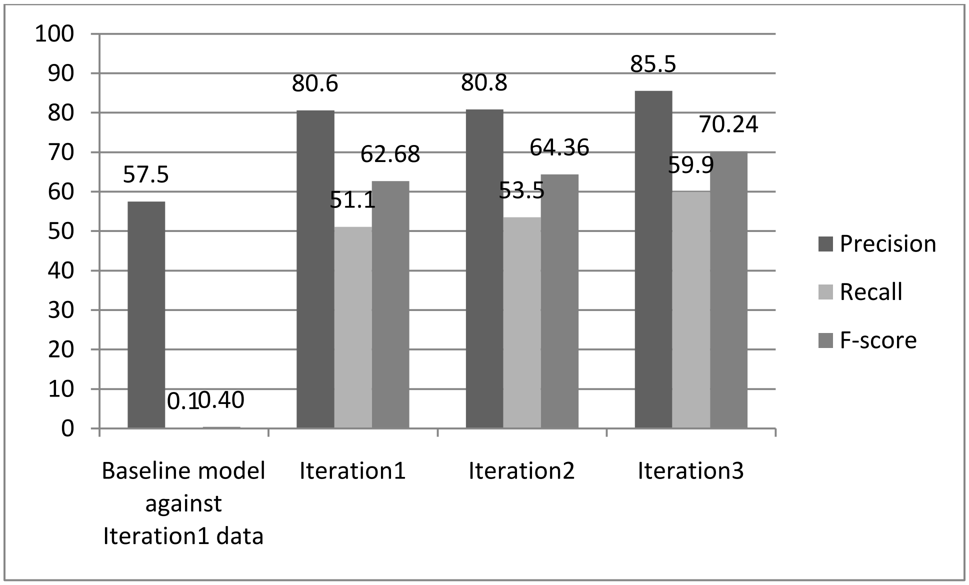

Figure 1. Entity Extraction Performance Metrics

3.2.2 Name Matching

To evaluate name matching performance, we use the standard precision and recall metrics from information retrieval. Initial testing was done on an Arabic subset of a large multicultural name test set to ensure that the matcher worked effectively on the kind of names most often queried by users. We then tested the matcher using the entire multicultural data set to observe how well it performed on non-Arabic queries. For the name matcher integrated into the system reported on in this paper, the best-performing precision and recall were 62 and 56, respectively, when tested on the entire data set, consisting of Arabic and non-Arabic queries. For more information on testing procedures and results, as well as additional improvements to the name search algorithm see (Arehart, 2010).

3.3 System-level Evaluation

The system level evaluation, adopted from similar work on name-matching evaluations (Arehart, 2008; Miller, 2008), was used to test whether the complete system would return the correct documents given a name query. An existing MITRE-developed evaluation infrastructure (Miller, 2008) was used to develop a TIGR-specific ground truth data set; customized metrics calculators were used to measure system performance on that data set. Since evaluation of every potential query name document match is impractical, the ground truth creation process was based on the methodology from National Institute for Standards in Technology (NIST) Text REtrieval Conference (TREC) (Voorhees and Harman, 2000; Voorhees, 2001). This bootstraps the process by querying the system and other baseline systems (with search parameters set to permissive thresholds), and collecting the results for human adjudication.

To create the ground truth set for the system evaluation we used documents which were not used to train the entity extraction model(s). The 1193 test queries include 264 names from the annotated target documents, 276 hand-created variants of those names, and 653 actual TIGR user queries. We indexed the names tagged in the human-annotated version of the target documents. Next we used the name matcher – set at a low return threshold - to compare the test queries to the indexed names and return the relevant documents. A group of human adjudicators reviewed the query-document matches to determine if the document actually did contain a name that matched the query. Those judgments were recorded and compiled into the ground truth set for the system evaluation.

When evaluating the updated TIGR system, we followed a similar procedure. We ran the original version of the target documents through the entity extraction model, and indexed those names that the model identified. Then, we used the optimized name matcher to compare the query list to the index of model-tagged names. We also ran the test queries and target documents through the original TIGR implementation to be able to measure any improvement in performance. The metrics calculators mentioned above compared the returned documents to the ground truth data set, and calculated the precision, recall, and F-score metrics for the original and updated TIGR systems. Those metrics are in Figure 2.

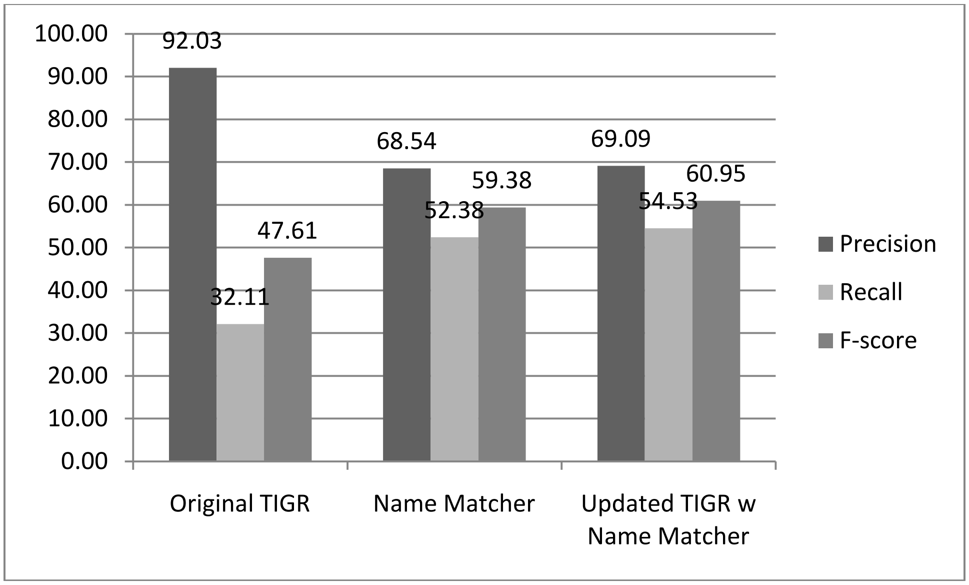

Figure 2. System-level Performance Metrics

4 Conclusions and Future Work

In this paper we presented the evaluation of updating an information retrieval system. The system update involved adding an entity extraction model to identify person names within free text documents to enable more advanced indexing, as well as updating the search capability to include an algorithm optimized for matching Romanized Arabic person names (Arehart, 2010). By using the best-performing versions of the entity extraction model and name matcher, we were able to improve search performance in the TIGR system from an F-score of 47.61 to 60.95, including an approximately 170% increase in recall. The improved search capability will help users better identify pertinent documents within the TIGR system.

Performing system-level evaluation in addition to component-level evaluation aided us in identifying other possible improvements to the TIGR search system. With the end-to-end, or system-level, evaluation infrastructure in place, we can easily evaluate how different entity extraction models or name matching algorithms affect overall system performance. For instance, we can investigate how additional document features can be used to train the entity extraction model and affect both component- and system-level performance.

Perhaps more interestingly, we discovered combined performance effects that are not evident in the component-level evaluations alone. That is, issues were exposed that are not problems for either the extraction or name matching components individually, but do present problems when the components are combined into an end-to-end document retrieval system. The most obvious of these effects is a result of the independence of the name extraction and indexing processes, which leads to imprecision in matching at the document level. That is, each name extracted from a document is indexed independently, and without reference to other representations of that name in the document.

<u>Document:</u> Salim Mohammed al Massri is the proprietor of Mid East Fine Foods. Salim moved to the United States in 1987, and since this time …. Later, Salim al Massri….

<u>Indexed names:</u> Salim Mohammed al Massri, Salim, Salim al Massri

<u>Query:</u> Salim bin Hassan Abd al Rahman

Figure 3. Sample Document/Index/Query

Thus, shortened representations of a name in the document index match a much wider set of queries than would be the case if the coreference chain were taken into account during the indexing process. In the example in Figure 3 above, the query Salim bin Hassan Abd al Rahman would (erroneously) match the sample document since the single index entry "Salim" refers to this document. This would not be detected as a false positive in the component-level evaluation of either the extraction process or of the name matcher, since the name "Salim" does occur in isolation in the document, and the name match of "Salim" to "Salim bin Hassan Abd al Rahman" would be considered a plausible match in the absence of other information. It is only in the system-level evaluation that this problem is detected. One piece of future work is thus incorporating coreference resolution within, and perhaps across, documents to determine how much that would increase search effectiveness. Another future enhancement currently under development is a query classification model, which would enable the system to route queries to the most appropriate type of search algorithm, such as a fuzzy name matcher or an exact matcher, a matcher optimized for a particular culture, or a standard IR engine without name search enhancements in the case of non-name queries.

References

Arehart, Mark and Keith J. Miller (2008). A Ground Truth Dataset for Matching Culturally Diverse Romanized Person Names. Language Resources and Evaluation Conf., Marrakech, Morocco.

Arehart, Mark (2010). Indexing Methods for Faster and More Effective Person Name Search. Language Resources and Evaluation Conf., Valletta, Malta

DARPA (2009). Advanced Soldier Sensor Information System and Technology (ASSIST). http://www.darpa.mil/ipto/programs/assist/assist_tigr.asp

Douthat, A. (1998) The Message Understanding Conference Scoring Software User's Manual. In Proceedings of the 7th Message Understanding Conference (MUC-7). http://www.itl.nist.gov/iaui/894.02/related_projects/muc/muc_sw/muc_sw_%manual.html.

Lafferty, J., A. McCallum, and F. Pereira (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Intl. Conf. on Machine Learning2001.*

Miller, Keith J., Mark Arehart, Catherine Ball, John Polk, Kenneth Samuel, Elizabeth Schroeder, Eva Vecchi and Chris Wolf (2008). An Infrastructure, Tools and Methodology for Evaluation of Multicultural Name Matching Systems. Language Resources and Evaluation Conf., Marrakech, Morocco.

Voorhees, E. M. (2001). The Philosophy of Information Retrieval Evaluation. Lecture Notes in Computer Science; Revised Papers from the Second Workshop of the Cross-Language Evaluation Forum on Evaluation of Cross-Language Information Retrieval Systems, 2406 (pp. 355-370). London, UK: Springer-Verlag.

Voorhees, E. M. and D. Harman (2000). Overview of the Eighth Text REtrieval Conference (TREC-8). In D. Harman, editor, The Eighth Text REtrieval Conference (TREC-8), Gaithersburg, MD, USA, 2000. U.S. Government Printing Office, Washington D.C.

Wellner, B. (2008). Carafe: ConditionAl RAndom Fields, Etc. http://sourceforge.net/projects/carafe/

Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. Proceedings of the Section on Survey Research Methods. American Statistical Association (pp. 354-359).