

Processing Named Entities in Text

*Paul McNamee, James C. Mayfield,
and Christine D. Piatko*

*U*nderstanding human communication is a key foundation on which the understanding of human dynamics is based. Detection and classification of names in text and resolving mentions of those names to real-world entities are language-understanding tasks that might reasonably be automated. The need for these techniques arises in numerous settings such as news monitoring, law enforcement, and national security. In this article we give an overview of research in the area, describe automated techniques used for identifying and relating names in text, and discuss community evaluations that have given a significant boost to research efforts worldwide. We also highlight APL's contributions to research into some of these problems, giving particular emphasis to a recent evaluation of algorithms to match entities in text against a large database.

INTRODUCTION

Five months after the Christmas Day 2009 bombing attempt on Northwest Airlines Flight 253, U.S. Customs and Border Protection officials removed convicted Times Square terrorist Faisal Shahzad from an airplane at John F. Kennedy airport after they reviewed the passenger manifest and found his name on the “No Fly” list. In this instance, checking names against a watch list proved effective. Three days later, a different Emirates Airlines flight was stopped because of a false match.¹ After a delay of more than an hour, the misidentified passenger was allowed to reboard, and the

plane departed. These incidents demonstrate both the value of effective name-matching technology and the cost of poorly performing algorithms.

Applications for technology that can resolve personal names include gathering census data, linking patient health records from separate hospitalizations, mail delivery, prevention of identity crimes, law enforcement (e.g., serving arrest warrants), and national security (e.g., border control and terrorist watch lists). It is an increasingly vital technology but one with real consequences for both false positives (predicting incorrect matches)

and false negatives (failing to detect a match). In the United States, an innocent person is arrested as a result of mistaken identity almost every day.^{2,3}

Automatically extracting information from text has been a recurring need for the U.S. military, and the Defense Advanced Research Projects Agency has conducted several programs to advance the technology.⁴ In 1995, the Sixth Conference on Message Understanding (MUC-6) was held to evaluate the performance of information-extraction systems. During the design of this evaluation, the phrase “named entity” was coined to describe textual references to real-world entities, and tasks were developed for automatically recognizing mentions of named entities (now known as “named entity recognition,” or NER) and linking coreferential mentions of the same entity (known as “coreference resolution”). Most research in this area has focused on coarse-grained entity types such as person, organization, and location.

One of the significant drivers of progress in human language technology over the past two decades has been the establishment of community-wide evaluations of individual technologies, such as MUC-6. These evaluations provide a common task definition, data sets, and evaluation metrics; any interested research group may participate. There are several advantages to this approach, including reducing the costs of obtaining and annotating data, enabling direct comparison of results because of the uniformity of the conditions and metrics, and providing a forum where researchers interested in a common problem can come together. Table 1 lists some of the major international evaluations of named-entity detection and classification. APL has participated in a number of these evaluations; we present some of our results in this article.

Table 1. Major international NER evaluations.

Evaluation Name	Year	Language(s)
MUC-6	1995	English
MUC-7	1997	English
CoNLL	2002	Dutch, Spanish
CoNLL	2003	English, German
ACE	2005	Arabic, Chinese, English
HAREM 1	2006	Portuguese
ACE	2007	Arabic, Chinese, English, Spanish
ACE	2008	Arabic, English
NERSSEAL	2008	Bengali, Hindi, Oriya, Telegu, Urdu
HAREM 2	2008	Portuguese

HAREM 1 and 2, Reconhecimento de Entidades Mencionadas Em Português 1 and 2; NERSSEAL, Workshop on NER for South and South East Asian Languages.

When attempting to match persons, organizations, and locations, exact-string matching alone is not a viable approach. False positives occur because distinct entities can share a name (name ambiguity). False negatives occur because different names can refer to the same entity (e.g., nicknames, aliases, or legal name changes) and because name variants can be nontrivial to match because acronyms, abbreviations, or foreign translations and transliterations (name variation) are used or because fragments are omitted.

Names of organizations can be particularly difficult to identify and match because of the pervasive use of high-ambiguity acronyms (for example, our own organization name, “APL,” might refer to a computer programming language, the political party Alliance pour le Progrès et la Liberté, the disease acute promyelocytic leukemia, etc.), atypical orthography (e.g., go!, 1-800 Contacts, accenture, eHarmony), and longer names that are routinely shortened (e.g., referring to the United States Centers for Disease Control and Prevention as Center for Disease Control).

Geographic places are often named after existing locations (e.g., New Amsterdam, York) or famous people (e.g., Pennsylvania named for William Penn, or Lincoln, Nebraska, named for Abraham Lincoln). Surnames, when indicative of ancestral origin, can derive from locations (e.g., Milano from Milan). Location names can also be part of organization names, such as in Tennessee Department of Correction.

When names are translated or transliterated from a foreign language, especially one with a different phonemic lexicon or one that uses a different writing system, multiple accepted variants can be formed. A well-known example is Libyan ruler Muammar al-Gaddafi, whose surname can be spelled in dozens of different ways. According to the *Christian Science Monitor*,⁵ the U.S. State Department spells it Qadhafi, the Associated Press uses Gadhafi, Reuters uses Gaddafi, the *Los Angeles Times* uses Kadafi, and the *New York Times* uses Qaddafi.

In short, names are complex, do not always follow normal rules for orthography, and may refer to a variety of entity types. Because new names can be created anytime, all names cannot be exhaustively enumerated. Thus, sophisticated algorithms are needed to identify and match names.

There are three chief problems in processing written names: (i) correctly recognizing the presence, extent, and type of names; (ii) linking the separate references to an entity within a single document; and (iii) identifying references to the same entity across multiple documents. We discuss each of these problems in the following sections and present some of our research in NER and entity linking.

RECOGNITION AND CLASSIFICATION OF NAMED ENTITIES

NER consists of identifying in a text sequences of words that correspond to a predefined taxonomy of entities, such as people, organizations, and locations. As with the related technology of part-of-speech tagging, most approaches to NER attempt to label each word in a sentence with its appropriate class. For part-of-speech tagging, these classes are syntactic classes, such as adjectives, prepositions, common nouns, etc. In NER, the taxonomy of entities is usually small, and nonentities are often given a separate “not-an-entity” tag. For this article, we focus on person (PER), organization (ORG), and location (LOC) entities and use the designation NIL to represent nonentities; however, there are many other types of entities that might be of interest, such as product names, titles of works of art, and types of vehicles. Table 2 gives some examples of names that can be difficult for automated systems to detect and label correctly.

Rule-Based Approaches to NER

Early work in NER (e.g., Ref. 6) examined rule-based approaches. For example, a rule for detecting person names might identify an honorific followed by one or more capitalized words (e.g., *Adm. Rickover* or *Dr. Lise Meitner*). Equation 1 is a regular expression illustrating this rule. Similarly, Eq. 2 identifies names of companies that consist of a series of capitalized words followed by an indicative suffix such as *Inc.*

$$(\text{Mr} \mid \text{Mrs} \mid \text{Ms} \mid \text{Dr} \mid \text{Adm}).? ([A - Z][a - z]^+) + \quad (1)$$

$$([A - Z][a - z]^+) + (\text{Inc} \mid \text{Co} \mid \text{Corp}).? \quad (2)$$

Rules such as these can be brittle and in practice can be significantly more complicated than those presented here. Equation 1 does not match person names that lack an honorific or that contain initials; Eq. 2

Table 2. Examples of names that pose challenges for automated NER systems.

Name	Type
I Can't Believe It's Not Butter	Product
Sunday	Nonentity
Billy Sunday	Person
Ohio	Location
USS <i>Ohio</i>	Vehicle
Attorney General	Nonentity
Royal Society for the Prevention of Cruelty to Animals	Organization
Harry S. Truman Presidential Library and Museum	Facility
World War II	Nonentity

does not correctly match organization names that contain a lowercase conjunction or preposition (e.g., *Bath and Body Works, LLC*). To be effective, patterns must be crafted with care and extensively tested. However, rule-based approaches have several advantages: First, unlike the statistical approaches discussed below, rule-based techniques do not depend on the availability of labeled training data, which can be expensive to obtain. Second, regular expressions require only a small amount of memory and can be implemented as finite-state recognizers, capable of running in time linear with the length of the input text.

Statistical Approaches to NER

In the 1990s the field of computational linguistics was undergoing a shift from knowledge-intensive paradigms (such as linguistic rules) to data-oriented approaches based on a combination of statistical learning and the large data sets that were enabled by advances in computer storage. NER was no exception to this trend. It is often easier for a person to identify that a given word should take a particular tag (e.g., “I know this is the name of a company”) than it is to articulate a rule for identifying words in that category. Consequently, there is significant interest in machine-learning approaches to tagging. Such approaches take as input a body of text that has been tagged in the desired manner and from such training data induce a mechanism for tagging new text. The dominant model today is based on statistical language modeling. In their simplest form, statistical models are based on estimating two probabilities:

1. $p(\text{word}_i \text{ hasTag } T_x)$ —the prior probability that a word belongs to a particular category, or tag. For example, $p(\text{Brown hasTag PER})$, the probability that the word *Brown* is part of a person's name, might be 0.40, while $p(\text{City hasTag LOC})$, the probability that the word *City* is part of a location name, might be 0.85.
2. $p(\text{word}_i \text{ hasTag } T_x \mid \text{word}_{i-1} \text{ hasTag } T_y)$ —the conditional probability that a word belongs to a particular category, given the category assigned to the preceding word. For example, $p(\text{Smith hasTag PER} \mid \text{Jane hasTag PER})$, the probability that a word (e.g., *Smith*) following a word in a person's name is also part of the same name, might be 0.65.

If these two probabilities can be estimated for every word in a text and for every category, then a probability can be assigned to each possible sequence of labels across the words of the text. Efficient algorithms exist to calculate the highest probability label sequence, which is then used to assign a label to each word. Figure 1 shows the beginning of a sentence labeled in this way. For a more detailed description of these approaches, consult Manning and Schütze's⁷ excellent text, which discusses

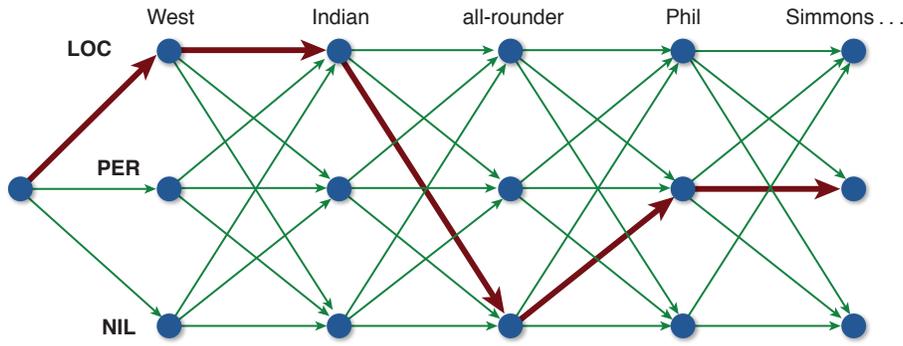


Figure 1. Lattice for NER. Each row represents a possible tag. Each column represents one word of the sentence to be tagged. A tagging corresponds to a path through the lattice.

this class of statistical models. Nadeau and Sekine⁸ provide an excellent overview of work in NER based on a review of more than 100 published studies.

In this approach, it is the role of machine learning to estimate each of the relevant probabilities. To do so, a machine-learning system must be able to distinguish the various uses of a word. A feature is a real value that represents some characteristic of the word. For example, one feature might indicate whether the word is capitalized, with 1.0 representing capitalized and 0.0 representing not capitalized. Another feature might indicate the percentage of time that this word appears as a noun in some large text collection. A third long-distance feature might indicate, for example, whether the word *president* appears within three words prior to the first occurrence of the word in the text. The vector of all such features of a word occurrence is used to represent that occurrence. The machine-learning algorithm uses these feature vectors for a training set of examples with known probabilities to learn how to assign probabilities to previously unseen feature vectors.

APL Innovations

The basic statistical method presented in the previous section works well when local features are predictive of output classes. However, it is difficult to make use of nonlocal features (features drawn from words that are beyond a small window around the word being assigned a label) within this framework.

Although a substantial amount of work has explored tagging tasks in English, other languages have been studied less. Language independence is difficult to achieve in tagging tasks because different languages appear to require different features. For example, some languages do not have cased letters, and thus proper nouns in those languages are not capitalized. Most taggers are severely limited in the number of features they may consider, because the computational expense of handling large numbers of features is high, and because the risk of overgeneralizing increases with the number

of features. Thus, the feature set must be finely tuned to be effective. Such constrained feature sets are naturally language dependent.

Increasing the number of features that a tagger can handle would ameliorate the language dependence problem, because the designer could select many relatively simple features in lieu of a few highly tuned features. It would also make possible the inclusion of arbitrary nonlocal features. APL's innovation was

to show how large numbers of features could be accommodated in this basic statistical framework.

Overtraining, or overfitting, is a well-known problem in statistical modeling where features found to be effective in training data are given excessive importance, which can lower performance when classifying subsequent data. Support vector machines (SVMs) can handle large numbers of parameters efficiently while simultaneously limiting overtraining.⁹ Thus, SVMs are well-suited for application to tagging. An SVM is a binary classifier that uses supervised training to predict whether a given vector belongs to a target class. All SVM training and test data occupy a single high-dimensional vector space. In its simplest form, training an SVM amounts to finding the hyperplane that separates the positive training samples from the negative samples by the largest possible margin. This hyperplane is then used to classify the test vectors; those that lie on one side of the hyperplane are classified as members of the positive class, whereas others are classified as members of the negative class. In addition to the classification decision, the SVM also produces a margin for each vector—its distance from the hyperplane. Figure 2 shows a sample SVM in two dimensions. Because SVMs do not produce probabilities,

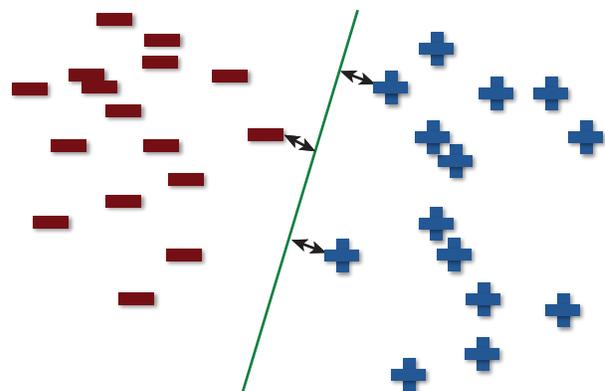


Figure 2. A simple SVM in two dimensions. A hyperplane (green line) separates the positive and negative examples. The vectors closest to the hyperplane are the support vectors.

we use the margin to estimate a probability; this method is ad hoc but produces good results in practice.

Experimental Results

The Conference on Natural Language Learning (CoNLL) sponsored evaluations for named entity tagging in 2002 and 2003. Texts in four languages were analyzed. Dutch and Spanish were studied in 2002, followed by English and German in 2003. We evaluated our approach to NER by using the CoNLL training and test sets. The evaluation metric is the *F*-measure, the harmonic mean of precision and recall, and it is defined as $F = 2PR/(P + R)$, where precision *P* is the percentage of strings identified by the system as named entities that are correct and recall *R* is the percentage of named entities actually present in the input that the system detects. Scores for each of the four languages are shown in Table 3. The results are quite good. APL's method, SVM-Lattice,¹⁰ outperforms each of the other methods we tested across all four languages. Had our best Spanish and Dutch runs been entered in the CoNLL-2002 evaluation, each would have placed second of 12 submissions (based on results reported by Tjong Kim Sang¹¹ and Tjong Kim Sang and De Meulder¹²). At CoNLL-2003, 16 teams participated, and APL's results were ranked fourth on the German task and seventh on the English task; the former result was within the published error bounds of the top system. Furthermore, we note that many of the competing systems used external language-specific resources, such as gazetteers and name lists, to improve their performance. With the exception of English, our results were accomplished with no knowledge of or tailoring to the language being tagged.

WITHIN-DOCUMENT COREFERENCE RESOLUTION

The goal of within-document coreference resolution is to identify all places where a given entity is mentioned in a single document. Especially in news, but also in other documents, one often finds an initial canonical reference to an entity (e.g., Coach Joe Gibbs) followed by additional referring expressions (e.g., Mr. Gibbs, the

Table 3. Performance on CoNLL data.

Language	Baseline System (%)	APL System (%)
Dutch	67.9	75.5
Spanish	72.3	80.8
English	75.5	84.7
German	59.0	70.0

The baseline system is a traditional Hidden Markov Model that looks at only local features (the current and previous words). Values are *F*-scores (harmonic mean of precision and recall).

coach, he, the three-time Super Bowl winner). Mentions can be classified as name, nominal (a common noun, such as president, coach, admiral, etc.), or pronominal (e.g., he). The goal of coreference resolution is to detect and link these multiple references into a single "coreference chain." A variety of statistical learning methods have been applied to this problem, including decision trees,¹³ maximum-entropy models,¹⁴ and graph partitioning.¹⁵ Typical features include mention similarity, lexical context of mentions, position in the document, and distance between references. Typically, these methods run in time proportional to the square of the number of mentions in the document; however, because most documents are short, this approach does not usually present difficulties.

Generally the surface forms of an entity mentioned in a document (i.e., the various words and phrases that refer to the entity) are unambiguous; that is, it is rare for a document to mention two entities that have the same name. An occasional exception to the rule can occur in text such as the following (which describes two men named Wes Moore¹⁶): Wes Moore, a Johns Hopkins graduate, Rhodes scholar, and former aide to Condoleezza Rice, was intrigued when he learned that another Wes Moore, his age and also from Baltimore, was wanted for killing a cop.

Because such instances are rare, solutions to the within-document coreference resolution problem usually focus on identifying and fusing name variants rather than on disambiguating names.

While rule-based heuristics can be helpful in identifying coreferent entities, they are often not as accurate as statistically trained classification techniques. For example, the heuristic of choosing the closest preceding plural noun as referent for the pronoun they or them leads to only about 55% precision.¹⁷ In contrast, machine-learning systems can obtain 84% precision in resolving pronouns.¹⁸

CROSS-DOCUMENT COREFERENCE RESOLUTION AND ENTITY LINKING

Unlike the single-document case, resolution of entities across multiple documents must directly address the name ambiguity problem. Two closely related problems have been addressed in the literature: cross-document coreference resolution and entity linking.

Cross-document coreference resolution is the problem of linking together all mentions of the same entity across multiple documents. Some researchers reduce this problem to the formation of document clusters. For example, given 100 documents that mention a *John Smith*, the documents are clustered based on which *John Smith* they appear to discuss. The Web People Search (WePS) workshops have adopted this model.¹⁹

In contrast, the 2008 Automated Content Extraction (ACE 2008) evaluation, which held a cross-document task in both Arabic and English, required individual mentions to be linked across documents. Names tend to follow a power-law distribution; the most frequently mentioned real-world entity usually occurs much more often than the 10th- or 20th-most frequent. As a result, clusters vary dramatically in size. In the ACE 2008 exercise, roughly 10,000 documents were provided in each language, and systems were required to produce appropriate clusters of entity mentions. A back-of-the-envelope calculation reveals that a brute force $O(n^2)$ process would require 10 billion comparisons, with each comparison potentially requiring substantial processing. A successful system must find some way to dramatically reduce the number of comparisons made.

Entity linking (also known as record linkage or entity resolution) is closely related to cross-document coreference resolution. In entity linking, a set of known entities is available, and the task is to determine whether an entity mentioned in text is one of these known entities, and if so, which one. Winkler²⁰ provides an overview of entity linking based on research at the U.S. Census Bureau. Because of its broad coverage and immense popularity, linking entities to Wikipedia has been a focus of several studies, including a community evaluation, the Text Analysis Conference Knowledge Base Population (TAC-KBP) exercise.²¹

Traditional approaches to cross-document coreference resolution have first constructed a vector space representation derived from local (or global) contexts of entity mentions in documents and then performed some form of clustering on these vectors.

APL Innovations

APL researchers participated in the ACE 2008 cross-document coreference resolution task and the TAC 2009 entity-linking task, as part of a team with The Johns Hopkins University (JHU) Human Language Technology Center of Excellence (HLTCOE).^{22–24} The TAC 2009 track made available a surrogate knowledge base (KB) by taking an October 2008 snapshot of Wikipedia and extracting pages that contained semistructured “infoboxes”—tables of attributes about the page’s subject. More than 818,000 entities were captured. A sample Wikipedia infobox is shown in Fig. 3. The entity-linking task consisted of taking a given news article and a name mention contained in the document and returning the correct KB identifier (if the entity is present in the KB) or the token NIL (if it is not present in the KB). Table 4 illustrates this task.

Approach

Our approach to entity linking proceeds in two phases. We start with a triage phase, in which we select

Virginia Woolf



Born	Adeline Virginia Stephen 25 January 1882 London, England, UK
Died	28 March 1941 (aged 59) near Lewes, East Sussex, England
Occupation	Novelist, Essayist, Publisher, Critic
Notable work(s)	To the Lighthouse , Mrs Dalloway , Orlando: A Biography , A Room of One's Own
Spouse(s)	Leonard Woolf (1912–1941)

Figure 3. Sample Wikipedia infobox.

a subset of KB entries that are reasonable candidates for the entity mention we are trying to link. This reduces the number of KB entries we must consider from nearly a million to a median of 16 and a maximum of a few thousand. Our goal in this first phase is to achieve high recall, i.e., to miss few correct entries in our candidate set while still dramatically reducing the number of entries we consider. To that end we considered a small number of fast-to-compute features based on simple string comparisons and known aliases.

In our TAC 2009 entry, this processing phase correctly included the appropriate candidate in the reduced entity set 98.6% of the time. Some of the difficult cases for which our system failed to include the correct KB node include: *Iron Lady*, which refers metaphorically to Yulia Tymoshenko; PCC, the Spanish-origin acronym

Table 4. Example of a KB entity-linking task.

No.	Name	Descriptor	Birth–Death
1	John Williams	Archbishop	1582–1650
2	J. Lloyd Williams	Botanist	1854–1945
3	John J. Williams	U.S. senator	1904–1988
4	John Williams	Author	1922–1994
5	Jonathan Williams	Poet	1929–
6	John Williams	Composer	1932–
7	John Williams	Politician	1955–

The linking task is illustrated using John Williams and the text “Richard Kaufman goes a long way back with **John Williams**. Trained as a classical violinist, Californian Kaufman started doing session work in the Hollywood studios in the 1970s. One of his movies was *Jaws*, with **Williams** conducting his score in recording sessions in 1975.” Given a passage of text and a designated mention, the entity-linking task is to assign the mention to an entry in a database or state that the entity is not present in the database. In the passage above, John Williams refers to the American composer well known for his Academy Award-winning film scores. A system presented with the query *John Williams* found in this text would be required to identify John Williams no. 6 as the correct target.

for the Cuban Communist Party; and *Queen City*, a former nickname for the city of Seattle, Washington.

Our second phase, the candidate ranking phase, ranks each of the candidate entries in the set of candidates coming from the triage phase by using supervised machine learning. As in our approach to NER, we represent each candidate KB entry as a vector of real-valued features. We use an SVM variant that supports ordinal regression called a ranking SVM²⁵ to rank each of the candidate KB nodes as well as the NIL response; we then select the top-ranked node as the system’s answer.

We developed approximately 200 features, which are the core of our approach. At the simplest level, features can be broken down into string comparison features (those principally based on the intrinsic properties of the names involved), document features (those based on comparisons between documents, i.e., between the text containing the query and the Wikipedia page from which the infobox was extracted), and other features. In the following discussion, *Q* is the query name (the name mentioned in the text for which we are trying to find the correct KB entry), *K* is the name of the KB node being considered, *K-Text* is any text found within *K*, and *D* is the query document, i.e., the document that contains *Q*.

String Comparison Features

Equality

Naturally, if the query name *Q* and KB node name *K* are identical, this is strong (albeit not definitive) evidence that *Q* should be linked to *K*. Another feature

assesses whether names are equivalent after some transformation. For example, *Baltimore* and *Baltimore City* are exact matches after removing a common location word such as city.

Approximate Matching

Christen²⁶ investigated a wide variety of individual string comparisons, many of which we incorporated as features. We used set membership comparisons based on the character *n*-grams in *Q* and *K* (a character *n*-gram is simply a sequence of *n* contiguous characters found in a text). Specifically we used short *n*-grams, such as bigrams, trigrams, and skip-bigrams²⁷ and computed Dice coefficients (i.e., twice the size of the set intersection divided by the sum of the sizes of the two sets). We also computed the left and right Hamming distances, which detect strong prefix/suffix matches; the Hamming distance is the number of mismatched characters from two aligned strings. The ratio of the recursive longest common substring to the shorter of *Q* or *K* is effective at handling some deletions or word reorderings (e.g., *John Adams* and *John Quincy Adams*, or *Li Gong* and *Gong Li*). This method works by finding the longest common substring (e.g., “Adams” in the first example) and removing it from each string, then recursively identifying the longest common substring from the residual pieces and stopping the recursion when the length of the common substring found is less than some constant (we used a length of 2). Finally, checking whether all the letters of *Q* are found in the same order in *K* can be indicative (e.g., *Univ. Maryland* would match *University of Maryland*).

Acronyms

The automatic identification of acronyms enables matches such as those between *MIT* and *Madras Institute of Technology* or *Ministry of Industry and Trade*.

Aliases

Many aliases or nicknames are nontrivial to guess. For example *LUV* is the stock symbol for *Southwest Airlines*, and *Ginger Spice* is a stage name of *Geri Halliwell*. Selecting the *Ginger Spice* page in Wikipedia will take you to the *Geri Halliwell* page. We mined such redirects to create multiple type-specific lists of known aliases.

Document Features

Entity Mentions

We used features based on presence of names, that is, whether *Q* was found in *K-Text*, or whether *K* was present in *D*. Additionally, we ran a named-entity tagger and relation finder, SERIF²⁸ to find entity mentions that were coreferent with *Q* in *D* and tested whether the nouns in those entity mentions were present in *K-Text*.

KB Facts

KB nodes contain infobox attributes (or facts); we tested whether the words of the facts were present in *D*, either close to a mention of *Q* or anywhere in the provided article.

Document Similarity

Q and *K-Text* were compared using a standard information retrieval measure, cosine similarity with term frequency-inverse document frequency (TF/IDF) weights,²⁹ and also using the Dice coefficient from sets of words.

Other Features

Entity Classification

Each Wikipedia page is manually assigned a set of classes (e.g., Actor, Scientist, Politician, NFL Player, etc.) by Wikipedia editors. We use the classes assigned to its underlying Wikipedia page to assign a type (person, organization, location, or other) to each KB node. We then compare the apparent type of an entity in *D* with the type stored in the KB.

Prominence

Although it may be a dangerous bias to prefer common entities, it seemed helpful to at least estimate measures of popularity. We did this in several ways. The first approach was based on intrinsic properties of the KB nodes. We associated with each KB node several graph-theoretic properties of its corresponding Wikipedia page, namely, the number of in-links, the number of out-links, and the page length (in bytes). These served as a rough gauge of popularity. We also submitted the query string to Google and used the rank of Wikipedia pages in the Google results as an attribute for their corresponding KB nodes.

Categorical Features

Wikipedia pages are often labeled with human- or machine-generated metadata consisting of keywords or categories in a domain-appropriate taxonomy. In a system called Wikitology, collaborators at the University of Maryland, Baltimore County, have investigated use of ontology terms obtained by exploiting the explicit category system in Wikipedia as well as relationships induced from the hyperlink graph among related Wikipedia pages.³⁰ Following this approach we computed top-ranked categories for *D* and compared them to the categories for *K-Text* by using cosine similarity.

Indications of Absence

Some features indicate whether it is unlikely that there is a matching KB node for *Q*. For example, if there are many candidates with strong name matches, then it

is likely that one of them is the correct one; conversely, if no node has high similarity between *K-Text* and *D*, this increases the chance that the entity is absent from the KB.

Experimental Results

The HLTCOE team submitted three runs for the National Institute of Standards and Technology TAC 2009 entity-linking task. We trained our models by using 1615 hand-annotated examples. The first run used our entire set of features; the second and third runs each removed several features, which gave slight improvements on our development data set.

Our approach performed well on the TAC 2009 task. All our scores are substantially above median. Our third run received the top score across all participants in the evaluation when weighting each target entity evenly.

We observed that organizations were more difficult to associate than people or locations, which we attribute to the greater variation and complexity in naming organizations and to the fact that they can be named after persons or locations.

Feature Effectiveness

Given the large number of features we used to train our models, a natural question is “Which features proved most useful for the task?” We performed two analyses. The first type, an additive study, starts with a set of baseline features and measures the change when adding each group of features. The initial feature set for this study was the subset of string similarity features used in the triage phase. Our second analysis was an ablative study; this type of study starts by using all of the features and measures the change when subtracting each feature group.

Table 5 shows the changes that occur when different groups of features are added to our baseline feature set. The baseline condition is not very effective at finding correct alignments when target entities are present in

Table 5. Additive analysis of sets of features.

Class	All (%)	Non-NIL (%)	NIL (%)
Baseline	72.6	46.2	92.5
Acronym	73.2	48.6	91.6
Alias	72.3	50.8	88.4
Facts	69.7	55.6	80.2
Named entities	76.6	71.8	80.2
NIL	73.0	48.8	91.2
Popularity	76.0	74.2	77.3
String	69.7	51.0	83.8
Text	73.1	70.0	77.8
Type	71.4	50.2	87.4

Table 6. Ablative analysis of sets of features.

Class	All (%)	Non-NIL (%)	NIL (%)
Acronym	79.7	70.5	86.6
Alias	80.3	73.3	85.5
Facts	78.7	72.1	83.8
Named entities	78.4	67.4	86.3
NIL	79.4	72.2	84.8
Popularity	75.7	65.2	83.6
String	78.7	75.4	81.3
Text	80.8	75.0	85.1
Type	78.4	69.6	85.1
No ablation	79.8	70.6	86.8

the KB; the non-NIL percentage is only 46.2%. Inclusion of features based on analysis of named entities, popularity measures (e.g., Google rankings), and text comparisons (e.g., Q and KB document similarities) provided the largest gains.

Table 6 reports accuracy when feature groups are removed from the full set of features. The overall changes are fairly small, roughly $\pm 1\%$; however, changes in non-NIL precision are larger, about $\pm 5\%$. The relatively small degree of change indicates that there is considerable redundancy in our large feature set. In several cases, performance would have been improved by removing features.

SUMMARY

Three main problems in processing named entities in text are identifying the presence, extent, and type of a named entity; finding other mentions of the entity in the same text; and linking the entity to either other documents or to a central repository of known entities. We have shown that a machine learning-based approach that uses a wide variety of features of the text is effective for the first and last of these problems. Improvements in technologies for processing named entities will have benefits for areas as diverse as medicine, law enforcement, and homeland security.

ACKNOWLEDGMENTS: We thank our many CoNLL-2003, ACE 2008, and TAC-KBP 2009 collaborators, including David Alexander, Bonnie Dorr, Mark Dredze, Markus Dreyer, Jason Eisner, Tamer Elsayed, Tim Finin, Clay Fink, Marjorie Freedman, Nimesh Garera, Adam Gerber, Saif Mohammad, Douglas Oard, Claudia Pearce, Delip Rao, Asad Sayeed, Zareen Syed, Ralph Weischedel, Tan Xu, and David Yarowsky. This work is supported, in part, by the JHU HLT/COE. We gratefully acknowledge the provision of the *Reuters Corpus Vol. 1: English Language, 1996-08-20 to 1997-08-19* by Reuters Limited.

REFERENCES

- Hays, T., "Official: Stopped Flight Was False No-Fly Match," *newsday.com*, <http://www.newsday.com/news/nation/official-stopped-ny-flight-was-false-no-fly-match-1.1898686> (6 May 2010).
- Lindsay, S., "Nursing Mom Taken from Car, Wrongly Arrested, Jailed," *TheDenverChannel.com*, <http://www.thedenverchannel.com/news/4897809/detail.html?subid=22100484&qs=1;bp=t> (25 Aug 2005).
- Chen, S., "Officer: You've Got the Wrong Person," *Cable News Network (CNN)*, http://articles.cnn.com/2010-02-15/justice/colorado.mistaken.identity.arrest_1_arrest-police-aclu?s=PM:CRIME (15 Feb 2010).
- Rowe, P., "Holy Grail of Data-Sifting Proves Elusive," *San Diego Union-Tribune* (7 Dec 2009).
- O'Carroll, E., "Gaddafi? Kaddafi? Qadhafi? How Do You Spell It?" *The Christian Science Monitor*, <http://www.csmonitor.com/Commentary/editors-blog/2009/0923/gaddafi-kaddafi-qadhafi-how-do-you-spell-it> (23 Sept 2009).
- Gaizauskas, R., Humphreys, K., Cunningham, H., and Wilks, Y., "University of Sheffield: Description of the LaSIE System as Used for MUC-6," in *Proc. Sixth Conf. on Message Understanding (MUC-6)*, Columbia, MD, pp. 207–220 (1995).
- Manning, C. D., and Schütze, H., *Foundations of Statistical Natural Language Processing*, MIT Press, Cambridge, MA, pp. 317–380 (1999).
- Nadeau, D., and Sekine, S., "A Survey of Named Entity Recognition and Classification," *Linguisticæ Investigationes* 30(1), 3–26 (2007).
- Vapnik, V. N., *The Nature of Statistical Learning Theory*, Springer-Verlag, New York (1995).
- Mayfield, J., McNamee, P., Piatko, C., and Pearce, C., "Lattice-Based Tagging Using Support Vector Machines," in *Proc. 12th International Conf. on Information Knowledge Management (CIKM '03)*, New York, NY, pp. 303–308 (2003).
- Tjong Kim Sang, E. F., "Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition," in *Proc. Conf. on Computational Natural Language Learning (CoNLL-2002)*, Taipei, Taiwan, pp. 155–158 (2002).
- Tjong Kim Sang, E. F., and De Meulder, F., "Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition," in *Proc. Conf. on Computational Natural Language Learning (CoNLL-2003)*, Edmonton, Canada, pp. 142–147 (2003).
- Ng, V., and Cardie, C., "Improving Machine Learning Approaches to Coreference Resolution," in *Proc. 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA, pp. 104–111 (2002).
- Florian, R., Hassan, H., Ittycheriah, A., Jing, H., Kambhatla, N., et al., "A Statistical Model For Multilingual Entity Detection and Tracking," in *Human Language Technology Conf./North American Chapter of the Association for Computational Linguistics Annual Meeting (NAACL/HLT 2004)*, Boston, MA, pp. 1–8 (2004).
- Nicolae, C., and Nicolae, G., "Bestcut: A Graph Algorithm for Coreference Resolution," in *Proc. 2006 Conf. on Empirical Methods in Natural Language Processing*, Sydney, Australia, pp. 275–283 (2006).
- Moore, W., *The Other Wes Moore: One Name, Two Fates*, Spiegel & Grau, New York (2010).
- Wooley, B. A., "Pronoun Resolution of 'They' and 'Them,'" in *Proc. 11th International Florida Artificial Intelligence Research Society Conf.*, Sanibel Island, FL, pp. 280–282 (1998).
- Ge, N., Hale, J., and Charniak, E., "A Statistical Approach to Anaphora Resolution," in *Proc. Sixth Workshop on Very Large Corpora*, Orlando, FL, pp. 161–171 (1998).
- Artiles, J., Sekine, S., and Gonzalo, J., "Web People Search: Results of the First Evaluation and the Plan for the Second," in *Proc. 17th International World Wide Web Conf. (WWW2008)*, Beijing, China, pp. 1071–1072 (2008).
- Winkler, W. E., *Overview of Record Linkage and Current Research Directions*, Research Report, Statistics No. 2006-02, Statistical Research Division, U.S. Bureau of the Census, Washington, DC (2006).
- McNamee, P., Dang, H. T., Simpson, H., Schone, P., and Strassel, S. M., "An Evaluation of Technologies for Knowledge Base Population," in *Proc. 7th International Conf. on Language Resources and Evaluation (LREC)*, Valletta, Malta, pp. 369–372 (2010).

- ²²Mayfield, J., Alexander, D., Dorr, B., Eisner, J., Elsayed, T., et al., "Cross-Document Coreference Resolution: A Key Technology for Learning by Reading," in *AAAI Spring Symp. on Learning by Reading and Learning to Read*, Stanford, CA, pp. 1–6 (2009).
- ²³McNamee, P., Dredze, M., Gerber, A., Garera, N., Finin, T., et al., "HLTCOE Approaches to Knowledge Base Population at TAC 2009," in *Proc. Text Analysis Conf. (TAC)*, Gaithersburg, MD, pp. 1–10 (2009).
- ²⁴Dredze, M., McNamee, P., Rao, D., Gerber, A., and Finin, T., "Entity Disambiguation for Knowledge Base Population," in *23rd International Conf. on Computational Linguistics (COLING 2010)*, Beijing, China, pp. 1–9 (2010).
- ²⁵Joachims, T., "Optimizing Search Engines Using Clickthrough Data," in *Proc. Eighth ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining*, Edmonton, Canada, pp. 133–142 (2002).
- ²⁶Christen, P., *A Comparison of Personal Name Matching: Techniques and Practical Issues*, Technical Report TR-CS-06-02, Australian National University (2006).
- ²⁷Zobel, J., and Dart, P., "Finding Approximate Matches in Large Lexicons," *Software: Pract. Experience* 25(3), 331–345 (1995).
- ²⁸Boschee, E., Weischedel, R., and Zamanian, A., "Automatic Information Extraction," in *Proc. First International Conf. on Intelligence Analysis*, McLean, VA, pp. 1–6 (2005).
- ²⁹Salton, G., and McGill, M. J., *Introduction to Modern Information Retrieval*, McGraw-Hill Companies, Columbus, OH (1983).
- ³⁰Syed, Z. S., Finin, T., and Joshi, A., "Wikipedia as an Ontology for Describing Documents," in *Proc. Second International Conf. on Weblogs Social Media*, Seattle, WA, pp. 136–144 (2008).

The Authors



Paul McNamee



James C. Mayfield



Christine D. Piatko

Paul McNamee, James C. Mayfield, and Christine D. Piatko are computer scientists in the System and Information Sciences Group of APL's Milton S. Eisenhower Research Center and members of the APL Principal Professional Staff. They have collaborated on a variety of projects in human language technologies, including APL's HAIRCUT information retrieval system. In 2007 they helped establish the JHU HLTCOE, a DoD-funded research center adjacent to the JHU Homewood campus. Dr. McNamee is a senior member of the Association for Computing Machinery, and he teaches as an adjunct

faculty member in the JHU Engineering for Professionals part-time program in computer science. He is currently detailed to the HLTCOE in Baltimore, where his research includes topics in multilingual information retrieval and information extraction. At the HLTCOE Dr. Mayfield leads a research program in KB population and oversees the work of more than a dozen researchers and graduate students. He also has an appointment with the JHU Computer Science Department as an Associate Research Professor. Before joining APL in 1996, Dr. Mayfield was Associate Professor of Computer Science at the University of Maryland, Baltimore County. Dr. Piatko recently returned to APL after spending several years at the HLTCOE where she co-led programs in KB population and language understanding. She has

an appointment with the JHU Computer Science Department as an Assistant Research Professor. She is currently leading a project in designing a software architecture for human language technologies, and she is serving as a technical advisor for the Knowledge Discovery and Dissemination Program of the Office of the Director of National Intelligence's Intelligence Advanced Research Projects Activity. For further information on the work reported here, contact Paul McNamee. His e-mail address is paul.mcnamee@jhuapl.edu.