Knowledge-Based Extraction of Named Entities

Jamie Callan and Teruko Mitamura
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-8213, USA
{callan, teruko}@cs.cmu.edu

ABSTRACT

The usual approach to named-entity detection is to learn extraction rules that rely on linguistic, syntactic, or document format patterns that are consistent across a set of documents. However, when there is no consistency among documents, it may be more effective to learn document-specific extraction rules.

This paper presents a knowledge-based approach to learning rules for named-entity extraction. Document-specific extraction rules are created using a generate-and-test paradigm and a database of known named-entities. Experimental results show that this approach is effective on Web documents that are difficult for the usual methods.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Linguistic processing*; I.2.6 [**Artificial Intelligence**]: Learning—*Knowledge acquisition*

General Terms

Algorithms

Keywords

Named-entity extraction

1. INTRODUCTION

Named-entity detection has been an important shallow natural language processing tool for at least a decade. Early systems consisted of heuristic pattern recognition rules that were designed to locate the names of companies (e.g., [9, 7]), dates, times, monetary amounts (e.g., [8]), and similar concepts in unstructured text. The patterns for these first systems were created manually, and were to some extent tuned for a particular corpus. In spite of their obvious limitations, these first named-entity detectors were useful for a variety of natural language processing and information retrieval tasks.

Research on named-entity detection gathered momentum in the 1990s, and was a key feature of the Message Understanding Conferences (e.g., [11]). Accuracy and generality improved considerably during this time, in part due to increasing reliance on machine learning algorithms to learn the patterns that identified named-entities of various types. This line of research culminated in the use of Hidden Markov Models (HMMs) to learn statistical language models over words and simple word features (e.g., capitalization), which produced very accurate named-entity recognition on a variety of text types [1, 2, 10].

Named-entity detection is now a well-accepted shallow natural language processing technique, and is used in applications as diverse as fact extraction, question answering, information retrieval, and text data mining. The best systems are very accurate on the information sources and genres for which they are trained.

Prior research on named-entity detection can be viewed as learning local patterns or local grammars that identify named-entities. It assumes that there are local patterns or local grammars that are generally applicable, at least within a particular information source. This assumption is valid in environments where text is written by professionals according to well-specified editorial policies, as is the case for newswire, government, and many other types of documents. However, documents published on the Web often violate this assumption.

Text data mining on the Web has become a hot topic in recent years because of the large volume of information that is freely available. The canonical Web mining example might be creating a large structured text database from unstructured Web pages, for example databases of job openings [4]. Web text mining projects such as these rely on a certain amount of consistency among Web pages, and this reliance is often rewarded. To the extent that Web documents use complete natural language or follow consistent document formats, traditional named-entity recognition techniques can be successful.

One of our Web datamining projects focuses on providing information about a scientific field, such as Computer Science, to people who are unfamiliar with the field. In order to gain basic knowledge of an unfamiliar field fairly quickly, informaion about major scientists in the field, organizations or institutions doing interesting research, conferences or workshops to attend, and publications to read first would be useful.

CIKM'02, November 4–9, 2002, McLean, Virginia, USA.

Rule ID	62
Parent	22
Syntax	body/table/tr/td/p/br/list[1]-comma
Applies	39 times

Figure 1: A candidate extraction rule.

One of our goals on the projects is to identify senior members and organizations within an academic community. How can they be identified? Publication history is an important source of evidence, and traditional techniques are effective for finding them. Participation on an editorial board or as an organizer of a conference often indicates great expertise in a subject area, and may be a more important source of evidence, but it poses a problem. Journal and conference Web pages don't follow any consistent format, and the key information often appears in tables or lists. Traditional techniques don't work well for this type of information.

Indeed, we would argue that much of the interesting information on the Web is presented in a list, tabular, or telegraphic format that limits the effectiveness of traditional techniques for named-entity detection. When the use of language and format is not consistent across Web pages, patterns learned from one set of Web pages will not apply reliably to other Web pages.

In this paper we present *KENE*, a new approach to named-entity detection which uses a knowledge-based approach for learning entity extraction rules. *KENE* detects and extracts author names and institution names from conference and workshop Web pages.

2. KENE

When traditional approaches to named-entity detection fail, it is often because there is insufficient consistency among documents, and because the named-entities appear in list, tabular, or other telegraphic form. In these situations it is necessary to learn document-specific extraction rules. Document-specific extraction rules are often considered impractical on the grounds that sufficient document-specific training data is difficult or impossible to obtain. However, sometimes a *small* amount of training data is available, and a small amount may be all that is required to generate rules that are used only within a single document.

KENE is a system that takes a generate-and-test approach to named-entity extraction from structured documents. The document is parsed to produce a set of candidate extraction rules. A candidate extraction rule is a parse-tree path (sequence of document markup tags and punctuation) that reaches some string. The rule is terminated by the next occurring markup tag or selected punctuation character. Rules that differ only by the index of a node sibling (e.g., the i'th and $i+1$'th elements of an itemized list) are merged. The strings extracted by each rule are looked up in a database of known entity names. A rule that extracts a specified number of known names is considered validated; the other strings it extracts are assumed to be names also, and are inserted into the database.

An example candidate extraction rule is shown in Figure 1. This rule specifies a path through a parse tree that ends with a comma-separated list; the rule extracts the second element of the list (index 1, using zero-based counting). The rule applies 39 times in the document. When the rule is applied to the document text shown in Figure 2, it extracts the organization names "University of the Andes", "University of North Dakota", and "Hokkaido University".

```
<body> ... <table> ... <tr> ... <td> ...
<p> <b> <font color="#000066">
INTERNATIONAL PROGRAM
COMMITTEE: </font> </b> <br>
J. Aguilar, University of the Andes, Venezuela<br>
S.E. Anderson, University of North Dakota, USA<br>
K. Araki, Hokkaido University, Japan<br>
... </td> ... </tr> ... </table> ... </body>
```

Figure 2: A fragment of document text.

This approach to finding named entities is a variant of a method used by Cohen and Fan for extracting information from lists [5]. In that work a set of named-entities (musicians and musical groups) was used as a query to a Web search engine. Using multiple named entities as a query for retrieving Web pages increased the chances of retrieving pages that i) contained the *specified* named entities (as opposed to a page containing the words in an unrelated context), and ii) contained lists that included *other* of named entities.

KENE tests how well the knowledge-based approach to named-entity extraction works when there is no control over page contents. In particular, there is no a priori knowledge about what, if any, known named entities occur in the page. We compensate for this lack of knowledge by providing access to a large database of known named-entities (See 3.1).

It would be simple to extend KENE's document representation to include a larger set of features, for example lexical or syntactic features. A larger feature set would require testing more rules against the database, which could be costly. However, there is usually not sufficient repetition of lexical and syntactic features within a single document to supply a sufficient amount of training data, so in most situations they are probably not worth the added cost.

KENE requires that a potential entity name found in a document exactly match a known entity name contained in its knowledge base. This restriction is reasonable for KENE's task of identifying the names of authors and organizations, because these tend to be used relatively consistently. Cohen and Fan used a vector-space matching algorithm for matching the names of musical groups, accepting matches that satisfied a similarity threshold [5]. Vector-space matching seemed unlikely to work well for people names, because they are so short. Exact match is a simpler solution that has been sufficient. However, we do allow KENE a small amount of domain-specific knowledge, for example, knowledge that "Univ." can match "University" in organization names, to improve its exact-match accuracy.

3. EXPERIMENTAL METHODOLOGY

In this section we describe the data, metrics, and method used to test KENE experimentally.

3.1 Data: Knowledge Base

The knowledge base was created from citations of scientific publications in Computer Science and from author home pages identified by the University of Trier's Home-PageSearch Web site [6]. At the time of the experiments

it contained 93,989 author names and 9,292 organization names. Simple heuristics were used to identify and conflate names that differed due to minor spelling errors, common abbreviations, or minor naming variations, but there remain many multiple entries for single individuals (e.g., "John Smith", "J. Smith", "John Q. Smith") and single organizations.

3.2 Data: Web Pages

Two sets of pages (training and test) were downloaded from the Web. The pages were chosen because they fell into one of three categories: i) listed a journal editorial board, ii) listed a conference program committee, or iii) listed a conference program (e.g., paper titles and author names).

The training pages were available for examination and used during system development. This set was augmented occasionally during development, to provide new pages for blind testing. This set eventually consisted of 19 pages.

The set of 26 testing pages was kept separate from the training data, and was not used or examined during system development.

3.3 Metrics

Three metrics were used to measure the accuracy of named-entity detectors: Precision, Recall, and F-Score. They are defined as:

$$P = \frac{ee}{ee + eo} \quad (1)$$

$$R = \frac{ee}{ee + oe} \quad (2)$$

$$F = \frac{2PR}{R + P} \quad (3)$$

where:

ee: number of entities identified correctly;

eo: number of strings mistakenly claimed to be entities; and

oe: number of entities not identified.

These are standard metrics for evaluating named-entity detection [3].

3.4 Baseline System

We compared KENE to IdentiFinder, a well-known and very effective commercial product based on Hidden Markov Models [1, 2]. IdentiFinder was supplied to us already trained on a large corpus of newswire and similarly well-written text. IdentiFinder was used as it came "out of the box", without any tuning for this task.

IdentiFinder is a configurable system. It could have been tuned for this task, by adding features that recognized document structure (e.g., HTML tags), and by training on the set of training documents we examined during KENE system development. However, IdentiFinder is based on a Hidden Markov Model with many parameters. We felt that training on so few documents, and documents with so few similarities to one another, would have reduced its accuracy, not improved it.

4. EXPERIMENTAL RESULTS

A set of experiments was conducted to determine the effectiveness of KENE and IdentiFinder at finding people and organization names mentioned in the test set of 26 conference Web pages.

In this initial test, the validation thresholds were set to 10 for person names and 6 for organization names; that is, a candidate person extraction rule was required to extract 10 known person names to be considered validated. These values were determined empirically from experiments on the training data. We revisit the issue of parameter settings in Section 4.1.

The results from this first experiment are summarized in Table 1. KENE was considerably more effective than IdentiFinder at finding people and organization names. The relative Precision and Recall differences between KENE and IdentiFinder were 37% and 102% for people names. The relative differences were −6% Precision and and 300% Recall for organization names.

Four of the Web pages in the test data do not contain any organizations (LNCS2043, NDRM2001, WEBDB2001, WUAUC01). The experimental methodology gives Precision, Recall, and F-Score values of 0.0 when no organizations exist in a file, and these zeros are reflected in the average values (i.e., the row labelled "Averages"). One might argue that this is an unfair penalty, and that the averages are an underestimate of the effectiveness of each algorithm when organization names are present.

The "Averages, Subset" row reports the average Precision, Recall, and F-Score values with these four pages removed from the test set. Removing these four pages has little effect on the averages reported for person name detection, but improves the averages reported for organization name detection by 15-20% for both programs. These higher values are a more accurate indication of the effectiveness of each program (i.e., there is little value in measuring Precision, Recall, and F-Score on pages that contain no organizations).

We conclude from this experiment that KENE provides high Recall of named entities on structured Web pages without a significant loss of Precision.

The first set of experimental results (Table 1) do not explain *why* KENE was effective in these tests. One hypothesis is that most of the named entities were already in KENE's database, and in which case the task might be considered relatively easy, and the generate-and-test paradigm for creating document-specific extraction rules might be considered to contribute little.

We tested this hypothesis by examining Precision and Recall on known and unknown named entities. A *known* named entity is one that was in KENE's database prior to examining the document. An *unknown* named entity is one that was not in KENE's database prior to examining the document. We conducted this test by filtering known (or unknown) results from the file prior to scoring. One consequence of this testing methodology is that the sum of Recall on the known and unknown sets should be approximately equal to Recall on the complete set (Table 1). The relationship is only approximate due to roundoff errors.

Tables 2 and 3 report the results. KENE's Precision and Recall values for known and unknown data were essentially identical for Person names, and were close for Organization names. The similar Precision values indicate that the extraction rules were about equally accurate for names that were and were not in its database. The similar Recall values for known and unknown names indicates that, on average, about half the extracted names were in KENE's database of

	KENE						IdentiFinder					
	Person			Organization			Person			Organization		
File Name	P	R	F	P	R	F	P	R	F	P	R	F
Agents2001	75	67	71	0	0	0	0	0	0	0	0	0
AI-ED2001	75	96	84	0	0	0	85	76	80	50	4	7
ASC2001	94	92	93	91	61	73	80	62	70	58	6	11
CCGrid2001	68	95	79	51	36	42	0	0	0	0	0	0
CHI2001	100	91	95	97	97	97	0	0	0	33	16	22
EDP-2001	100	100	100	0	0	0	0	0	0	0	0	0
EHCI2001	38	100	55	72	93	81	92	56	70	62	33	43
ICAIL2001	81	96	88	0	0	0	93	84	88	100	8	15
ICSE2001	84	44	58	41	61	49	12	6	8	73	38	50
IP-Telephony2001	56	98	71	8	13	10	95	81	87	88	59	71
ISPD2001	35	38	36	0	0	0	31	31	31	18	10	13
ITACCESS01	0	0	0	44	33	38	2	1	1	17	9	12
IV2001	0	0	0	0	0	0	100	38	55	73	42	53
LNCS2043 *	97	52	68	0	0	0	0	0	0	0	0	0
MECO2001	64	100	78	87	75	81	59	57	58	16	4	6
MobiDE01	75	95	84	65	77	70	0	0	0	6	3	4
MS2001	93	100	96	94	82	88	73	64	68	21	2	4
NAACL2001	62	94	75	0	0	0	91	66	77	75	10	18
NDRM2001 *	100	100	100	0	0	0	81	85	83	0	0	0
RTAS2001	59	71	64	54	78	64	1	1	1	0	0	0
SMI2001	80	88	84	0	0	0	81	39	53	100	12	21
UIDIS2001	0	0	0	0	0	0	97	71	82	41	19	26
WEBDB2001 *	0	0	0	0	0	0	0	0	0	0	0	0
WUAUC01 *	100	100	100	0	0	0	86	65	74	0	0	0
WWW10	89	97	93	88	100	94	90	69	78	60	7	13
XP2001	83	83	83	31	42	36	9	8	8	2	1	1
Average	66	73	68	32	33	32	48	37	41	34	11	15
Average, Subset	64	75	68	37	39	37	50	37	42	41	13	18

Table 1: Effectiveness of two named entity finding strategies, as measured by the ability to find person and organization names in conference Web pages. The "Average" figures are computed over the complete set of 26 test pages. The "Average, subset" figures exclude the four test pages that do not contain any organizations (indicated by asterisks).

	KENE		IdentiFinder	
People Names	P	R	P	R
Average, Known	66	37	53	20
Average, Unknown	65	37	47	19

Table 2: Effectiveness of two named entity finding strategies, as measured by the ability to find known and unknown person names in conference Web pages. Evaluated over the 26 test Web pages that contain person names.

	KENE		IdentiFinder	
Organization Names	P	R	P	R
Average, Known	40	21	43	6
Average, Unknown	35	19	37	8

Table 3: Effectiveness of two named entity finding strategies, as measured by the ability to find known and unknown organization names in conference Web pages. Evaluated over the 22 test Web pages that contain organization names.

known named entities, and about half were new names.

Identifinder's Precision values are slightly higher for known names than unknown names (Tables 2 and 3), which is surprising. We assume that this result is due to an unusually skewed distribution of random errors across the two classes of named entities. The Recall values are approximately equal, which is what we would expect.

4.1 Parameter Settings

The experimental results reported in Tables 1-3 are based on validation thresholds of 10 for person names and 6 for organization names. These values were determined empirically based on the training data. They are higher than we expected initially. Lower thresholds might suffice if documents were consistently accurate in their use of markup, but Web documents frequently contain many markup errors, even when the document is generated automatically. We found greater variation in the markup of person names than in the markup of organization names, hence the higher thresholds.

We conducted a second set of experiments designed to study KENE's sensitivity to validation threshold values. Higher thresholds make it more difficult to validate rules; one would expect fewer rules to be validated, hence named-

Threshold	Pages	Average P	R	F
5	25	79	79	76
6	24	82	78	78
7	24	81	80	78
8	23	80	82	79
9	22	79	84	80
10*	22	78	86	80
11	21	76	86	79
12	21	76	86	78
13	21	76	84	78
14	20	76	85	78
15	19	78	89	81

Table 4: Effect of the validation threshold on the generate-and-test approach to named entity finding strategies, as measured by the ability to find people names in conference Web pages. The default setting for the other experiments is indicated with an asterisk (*). Evaluated over the test Web pages in which person names were found.

entities would be extracted for fewer pages. One might also expect that the rules validated by higher thresholds would be more Precise, because they extract a larger set of known names.

The effect of different validation thresholds cannot be determined accurately by measuring Precision, Recall and F-Score across the entire set of test pages. When a threshold is sufficiently high that no rule is validated, Precision, Recall and F-Score are zero; higher validation thresholds would be expected to produce more zero measurements, and hence *lower* Precision, Recall, and F-Score values. We compensated for this effect by measuring Precision, Recall, and F-Score only on pages for which at least one named-entity was extracted. We also report the number pages for which named-entities are extracted. One consequence of this change in experimental methodology is that the results are not directly comparable with Tables 1-3.

Tables 4 (for people names) and 5 (for organization names) show the effects of varying the validation thresholds used to test candidate extraction rules against the database of known named entities. The rows labeled with asterisks ('*') indicate the values corresponding to the experimental conditions reported in Tables 1-3.[1]

Varying the validation thresholds had only a small effect on Precision, Recall, and F-Score in these tests, indicating that the generate-and-test approach is relatively robust with respect to its threshold settings. In these experiments the validation thresholds learned from training data transferred reasonably well to the test data. A lower threshold for organization names would have increased scores slightly, primarily by validating rules for several additional Web pages.

In general, the validation threshold was inversely correlated with the number of pages for which named entities were extracted, which is not surprising. As the validation threshold was increased, fewer rules were validated.

Good thresholds for organization names tended to be smaller than good thresholds for people names in these experiments. This might seem surprising, but it is a consequence of the way these two types of names were used in these Web pages. People names were far more common than oranization names; and the use of people names was more varied (e.g., as author, reviewer, organizer, session chair, etc). Low validation thresholds for person name rules tended to produce a larger number of "noisy" rules. This was less of a problem for organization names, because they appeared less often, and when they appeared, they were used more consistently.

[1]The Precision, Recall, and F-Score values are higher in Tables 4 and 5 because there is no penalty when no extraction rule is validated for a Web page.

Threshold	Pages	Average P	R	F
3	16	64	69	65
4	14	65	67	65
5	14	63	66	64
6*	13	63	65	63
7	13	61	65	62
8	12	63	68	64
9	12	63	68	64

Table 5: Effect of the validation threshold on the generate-and-test approach to named entity finding strategies, as measured by the ability to find organization names in conference Web pages. The default setting for the other experiments is indicated with an asterisk (*). Evaluated over the test Web pages in which organization names were found.

4.2 Failure Analysis

One might expect that the generate-and-test approach would produce very accurate results, but the experimental results reported here (Table 1) indicate that a significant amount of error remains. A failure analysis indicates four common sources of error for the generate-and-test approach.

HTML errors: The documents in our test set contained many HTML errors. HTML errors sometimes allowed a validated (and otherwise-accurate) rule to extract text that was not a name.

Not enough known names: A rule is validated when it extracts a minimum number of known names. Occasionally no extraction rule was validated, either because the document was short, or because it contained a large percentage of unknown names.

Inability to generalize: KENE recognizes repeating structures in table and list elements identified by markup (e.g., <tr>, <li>), and it recognizes some types of repeating punctuation, because these are identified easily in a parse-tree. KENE does not consider every pair of sibling subtrees for possible repeating structure, so it failed to produce sufficiently general rules for tables and lists that were only implicit.

Insufficient structure: Markup tags and punctuation do not always provide enough information to define accurate named-entity extraction rules. For example:

- 10:00 John Smith

- John Smith Mary Jones

KENE was unable to generate valid extraction rules for these cases.

Inconsistent use of structure: The generate-and-test approach relies on consistent use of markup, but Web documents often use markup inconsistently. This was especially a problem when different sequences of markup produced the same image.

5. CONCLUSIONS

Named-entity detection is usually treated as a task of finding language patterns that are consistent across documents. That approach is successful when documents follow a consistent style, but it is less successful on Web documents that present information in telegraphic, tabular, or list form.

This paper presents an alternate approach in which document-specific extraction rules are generated and tested. Although there are no a priori constraints on what features to use in document-specific extraction rules, we focused on document markup and punctuation features so as to maximize the generality of the training data. A large database of known named-entities provided a method of testing and validating candidate rules.

The document-specific generate-and-test approach to finding named-entities is not meant to replace existing approaches to named-entity detection. It is clearly limited to environments, such as the Web or XML documents, where there is considerable document structure. We would expect it to be useful primarily for extracting named-entites from lists, tables, and portions of documents that rely heavily on visual structure (e.g., fonts and font characteristics).

Our experiments show that this approach to finding named-entities can in some cases provide considerably higher Recall than conventional methods, without a large drop in Precision. In one of our experiments, Precision was also increased; in the other, it did not.

A natural extension of the research reported here is to combine IdentiFinder's corpus-based language-pattern approach with KENE's document-specific markup-pattern approach. Data fusion often improves accuracy, particularly when the methods involved are of comparable accuracy but based on distinct forms of evidence, as are IdentiFinder and KENE. A tighter integration might strengthen both approaches. IdentiFinder might provide a path to adjusting KENE's validation thresholds dynamically, for example lowering them when the two approaches agree on a name, or raising them when they disagree. KENE might provide a source of relatively reliable training data for tailoring IdentiFinder's language patterns to individual documents. In general the combination of corpus-specific language patterns and document-specific markup patterns offers interesting research possibilities for improving named entity extraction.

Acknowledgements

We thank Maan Bsat, William Morgan, and Roger Braunstein for their assistance with this research. We also thank BBN for providing the IdentiFinder software.

This material is based on work supported by NSF grant IIS-9873009. Any opinions, findings, conclusions or recommendations expressed in this paper are the authors', and do not necessarily reflect those of the sponsors.

6. REFERENCES

[1] D. M. Bikel, S. Miller, R. Schwartz, and R. M. Weischedel. Nymble: A high-performance learning name-finder. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 194–201, 1997.

[2] D. M. Bikel, R. Schwartz, and R. M. Weischedel. An algorithm that learns what's in a name. *Machine Learning Journal*, 34:211–231, 1999.

[3] N. Chincor. MUC-7 named entity task definition dry run version, version 3.5. In *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Morgan Kaufmann, 1998.

[4] W. Cohen, A. McCallum, and D. Quass. Learning to understand the Web. *IEEE Data Engineering Bulletin*, 23(3):17–24, 2000.

[5] W. W. Cohen and W. Fan. Web-collaborative filtering: Recommending music by crawling the Web. In *Ninth International World Wide Web Conference*, 2000.

[6] G. Hoff. *HomePageSearch*. http://hpsearch.uni-trier.de/. University of Trier, 2002.

[7] T. Kitani and T. Mitamura. An accurate morphological analysis and proper name identification for Japanese text processing. *Journal of Information Processing Society of Japan*, 35(3):404–413, 1994.

[8] M. L. Mauldin. *Information Retrieval by Text Skimming*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1989.

[9] L. F. Rau. Extracting company names from text. In *Proceedings of the Sixth IEEE Conference on Artificial Intelligence Applications*, 1991.

[10] K. Seymore, A. McCallum, and R. Rosenfeld. Learning hidden markov model structure for information extraction. In *AAAI'99 Workshop on Machine Learning for Information Extraction*, 1999.

[11] B. Sundheim, editor. *Proceedings of the Third Message Understanding Evaluation and Conference*. Morgan Kaufmann, Los Altos, CA, 1991.