# Rule-based Word Clustering for Document Metadata Extraction

Hui Han[1,2]
[1]Yahoo Inc.
701 First Avenue
Sunnyvale, CA, 95129
huihan@yahoo-inc.com

Eren Manavoglu[2]
[2]Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA, 16802
manavogl@cse.psu.edu

Hongyuan Zha[2]
[2]Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA, 16802
zha@cse.psu.edu

Kostas Tsioutsiouliklis[1]
[1]Yahoo Inc.
701 First Avenue
Sunnyvale, CA, 95129
kostas@yahoo-inc.com

C. Lee Giles[2,3]
[3]School of Information Sciences and Technology
The Pennsylvania State University
University Park, PA, 16802
giles@ist.psu.edu

Xiangmin Zhang[4]
[4]Department of Library and Information Sciences
Rutgers University
4 Huntington, New Brunswick, NJ 08901
xzhang@scils.rutgers.edu

## ABSTRACT

Text classification is still an important problem for unlabeled text; CiteSeer, a computer science document search engine, uses automatic text classification methods for document indexing. Text classification uses a document's original text words as the primary feature representation. However, such representation usually comes with high dimensionality and feature sparseness. Word clustering is an effective approach to reduce feature dimensionality and feature sparseness, and improve text classification performance. This paper introduces a domain Rule-based word clustering method for cluster feature representation. The clusters are formed from various domain databases and the word orthographic properties. Besides significant dimensionality reduction, such cluster feature representations show a 6.6% absolute improvement on average on classification performance of document header lines and a 8.4% absolute improvement on the overall accuracy of bibliographic fields extraction, in contrast to feature representation just based on the original text words. Our word clustering even outperforms the distributional word clustering in the context of document metadata extraction.

## Categories and Subject Descriptors

H.4 [**Information Systems**]: Information Search and Retrieval-Clustering

## Keywords

Word Clustering, Feature Dimensionality Reduction


*SAC'05* March 13-17, 2005, Santa Fe, New Mexico, USA


## 1. INTRODUCTION AND RELATED WORK

Automatic document metadata extraction has been motivated by building unified services for heterogeneous digital libraries, enabling sophisticated querying of the databases and facilitating the implementation of the semantic web. **Document metadata** refers to the metadata from the **document header** (the text before the "introduction" or the end of the first page) and the **bibliographic fields**; document here means research papers. There has been much previous work in document metadata extraction e.g., Seymore et al. using hidden Markov models (HMM) for the document header metadata extraction [15]; Takasu using dual variable length output hidden Markov models for bibliographic fields extraction [17], etc.

The above methods use the original document text words for feature representation. These "bag-of-words" methods use statistical methods to train classifiers based on the words statistics. A drawback of such representations is that they usually have high feature dimensionality and/or feature sparseness, which makes computation expensive and can affect classification performance. One reason is that such features may be overly specific [3], e.g. name words "Mary", "Johnson" or "Tom".

There have been many successful feature dimensionality reduction methods: Latent Semantic Indexing (LSI) [4] and its probabilistic version (PLSI) [8] map the documents and words to a low dimensional latent semantic space. Feature selection methods choose useful features by thresholding based on the computation of information gain, document frequency, Chi square, etc. [19]. Word clustering methods cluster similar words, i.e. words in the same syntactic or semantic categories, and use the cluster labels as features for text classification. Word clustering reduces not only feature dimensionality, but also feature sparseness. Word clustering generalizes specific features by considering the common characteristics and ignoring the specific characteristics among the individual features. Distributional word clustering [12, 1, 16, 5], , a representative word clustering method, shows significant performance gain on text classification, and outperforms LSI and PLSI.

This paper introduces a method to cluster words according to the document syntactic structure, such as title, author, abstract, etc.

By using the clusters as features, we will have more features representative of the target class and that are similar to the metadata to be extracted. This idea is similar to Lin et al.'s CBC (Clustering by Committee) clustering algorithm [9], where the committee is a subset of the cluster members, and determines which other elements belong to the cluster. The features of the committee centroid tend to be the more typical features of the target class.

Our word clustering is based on **domain databases** and **word orthographic properties** [14], which contain a priori knowledge of a specific class. "**Domain**" corresponds to the class in text classification tasks. A **domain database** can be a name word database for the "author" class. Specific words are clustered based on their membership in the domain databases. For example, words "Mary", "Johnson" and "Tom", which appear in the name word database, are clustered and represented as ":name word:", the cluster label. Similarly, "Massachusetts" is represented as ":state:". We call this type of feature representation the *cluster feature representation*.

Word orthographic properties consider cases of the words, and digits or special characters the words contain. A **word** is a consecutive sequence of characters. "@" character is an orthographic property of the email address, and is used to cluster the specific email addresses as ":email:". Five-digit numbers are clustered and represented as ":digit[5]:". Such word orthographic properties have been effectively used in previous text processing tasks [2, 15, 3].

Our word clustering method appears to have low computational cost, and shows significant improvement on the performance of document header line classification and bibliographic field extraction, which is part of the document metadata extraction task. CiteSeer's performance rests on such algorithms.

This paper is organized as follows. Section 2 briefly introduces the related theories of expected entropy loss, hidden Markov models and Support Vector Machines. Section 3 describes our method of rule-based word clustering. Section 4 reports experiments on studying the effect of rule-based word clustering on classifying the lines of document headers and extracting bibliographic fields. Section 5 concludes and discusses the rule-based word clustering.

# 2. BACKGROUND

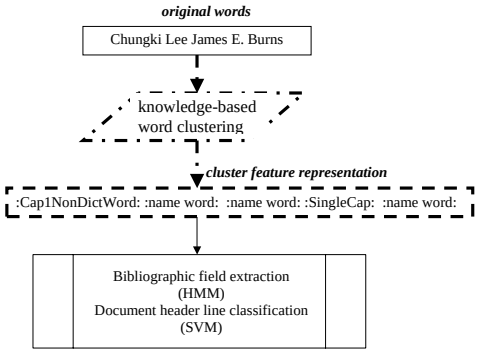


**Figure 1: A flow chart of word clustering and metadata extraction.**

Our word clustering is a piece of work on feature representation, which is a step before metadata extraction (Figure 1). The original text words are clustered and replaced by the cluster labels before further text processing. Next we describe briefly document metadata, a measure of feature ranking (expected entropy loss), and the methods used for metadata extraction (Support Vector Machines (SVM) and hidden Markov models (HMM)).

## 2.1 Document metadata

Metadata from document headers contains "title", "author", "affiliation", "address", "note", "email", "data", "abstract", "phone", "keyword", "web", "degree" and "pubnum". "Note" are the phrases about acknowledgment, copyright, notices, and citations; "degree" refers to the language associated with the thesis; "pubnum" means the publication number. The bibliographic fields contain "author", "book title", "date", "editor", "institution", "journal", "location", "note", "pages", "publisher", "tech", "title", and "volume" [15].

## 2.2 Expected entropy loss

Expected entropy loss [6] is synonymous with expected information gain. Entropy is computed independently for each feature. Let C be the event indicating whether the sample is a member of the specified class (A sample in this paper is a piece of metadata). Let f denote the event that the sample contains the specified feature (e.g., a sample of an address contains the feature word "avenue"). The prior entropy of the class distribution is $e \equiv -\Pr(C)\lg\Pr(C) - \Pr(\overline{C})\lg\Pr(\overline{C})$. The posterior entropy of the class when the feature is present is $e_f \equiv -\Pr(C|f)\lg\Pr(C|f) - \Pr(\overline{C}|f)\lg\Pr(\overline{C}|f)$; likewise, the posterior entropy of the class when the feature is absent is $e_{\overline{f}} \equiv -\Pr(C|\overline{f})\lg\Pr(C|\overline{f}) - \Pr(\overline{C}|\overline{f})\lg\Pr(\overline{C}|\overline{f})$. Thus, the expected posterior entropy is $e_f\Pr(f) + e_{\overline{f}}\Pr(\overline{f})$, and the expected entropy loss is $e - (e_f\Pr(f) + e_{\overline{f}}\Pr(\overline{f}))$.

## 2.3 Support Vector Machine classification

A SVM [18] attempts to find an optimal separating hyperplane to maximally separate two classes of training samples. SVMs are known for good generalization performance and ability in handling high dimensional data. In our task of document header metadata extraction [7], we use SVMs for classifying lines of a document header. While each line is represented by a vector of words the line contains, our word clustering algorithm transforms the original text word to a cluster label, and improves classification results.

## 2.4 Hidden Markov models

We apply HMMs [13] to bibliographic field extraction, and construct the HMMs for references[15] as follows: Each state corresponds to a bibliographic class, e.g. "author" and "title"; each word is an observation, and each state emits words following a class-specific multinomial distribution[10]. Extracting bibliographic fields from the unseen references using HMMs reveals the most likely state sequence for the observation, based on the transition probabilities and emission distributions learned from the training data.

# 3. RULE-BASED WORD CLUSTERING METHOD

Rule-based word clustering consists of the following three steps.

**Step 1: Generate domain databases.** We define two types of domain databases according to the way they are generated: the **External Domain Databases** and the **Constructed Domain Databases**. External Domain databases were collected from World Fact Book [1]; lists of 8441 first names and 19613 last names and Chinese last names, and standard on-line dictionary of linux system. These resources form the database of U.S. city names and major city names in other countries (city), the database of U.S. state names (state), the database of U.S. postcodes, e.g., "MA" or "NJ" , the database of country names (country), the database of name words, and the database of English words. Gazetteer [2] can also be used.

For classes without an available external domain database, e.g., "note" class and "pubnum" class, we construct domain databases from the positive training samples using Document Frequency (DF) thresholding [19]. The top-ranked words constitute the constructed domain databases. Table 1 shows top-ranked words for class "affiliation", "note", "pubnum" and "phone". The constructed domain

[1] http://www.cia.gov/cia/publications/factbook/
[2] http://www.census.gov/cgi-bin/gazetteer

databases provide complementary information even for the classes that have external domain databases, e.g. the address class.

| Affiliation | | Note | | Pubnum | | Phone | |
|---|---|---|---|---|---|---|---|
| DF | Feature | DF | Feature | DF | Feature | DF | Feature |
| 412 | university | 111 | research | 54 | report | 22 | fax |
| 111 | univ | 88 | support | 51 | technical | 17 | tel |
| 262 | department | 74 | grant | 29 | tr | 12 | phone |
| 103 | institute | 64 | science | 6 | csrp | 3 | usa |
| 62 | laboratory | 52 | part | 4 | memo | | |

**Table 1: Top words ranked by document frequency in four classes.**

**Step 2: Cluster Design.** We design clusters based on the domain databases and the words' orthographic properties. For example, words with mixed letters and digits and different cases may form the cluster ":Digs[2]::Capwords[2]::Digs[3]". Generally, each domain database corresponds to a cluster.

**Step 3: Rule Design.** We design rules to match words from different domain databases, check word orthographic properties, and then assign the word to an appropriate cluster. The rules consider multiple properties of the word to determine its cluster. For example, the word has to begin with an upper case letter and be in the name word database, before being assigned to the ":name-word:" cluster. The rules also address database conflicts in three ways in the cases when a word belongs to multiple domain databases as follows. **First,** we follow a "specific-to-general" order [11] to match the words with different domain databases. We design the following priority order of the domain databases in the header line classification experiment: $Postcode > Abstract > Keyword > Phone > Month > Addr > City > State > Country > Nameword > Word\ dictionary$. If a word appears in both the name word database and the standard linux word dictionary, it is assigned to the ":name-word:" cluster instead of the ":dict-word:" cluster. **Second,** we encode the multi-database membership of a word using a N-digit code. N is the number of databases the word belongs to. For example, for document headers, the "degree" database, "pubnum" database, "note" database and "affiliation" database have overlapping words. We use a four-digit binary code to indicate a word's membership in the above four databases. For example, ":1001:" means a word appearing in both "degree" and "affiliation" databases, but not appearing in the other two domain databases, "pubnum" and "note". **Third,** a word with multi-database membership forms an independent cluster by itself, i.e. keeps its original word format. This alleviates over-generalization of features. Replacing all digits by the feature ":number:" results in a lack of primitives for other features.

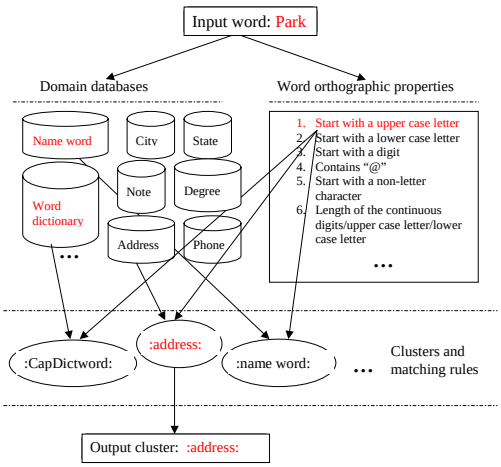


**Figure 2: An example of cluster assignment for the word "Park".**

Figure 2 shows an example of cluster assignment for the word "Park" . "Park" starts with the upper case letter and is in the address database, name word database and dictionary. According to the priority order of databases, we assign it to the "address" cluster.

# 4. EXPERIMENTS AND RESULTS

## 4.1 Experimental setup

We conduct two experiments: *document header line classification* and *bibliographic field extraction*. Document header line classification is the first step of our document header metadata extraction method. We train 15 classifiers based on the training data and classify each test line into the classes with positive classification scores. When all the classification scores are negative, we assign the test line to the closest class. Based on the line classification, the second step segments the lines of multiple class labels to further extract metadata, and recognizes individual authors from the multi-author lines. The performance of the first step line classification is critical to the performance of the following steps. We choose Support Vector Machines as the method for line classification, and hidden Markov models for bibliographic fields extraction.

In both experiments, we try three different types of feature representations: the original text words, the cluster features by Baker et al.'s distributional word clustering method, and the cluster features of our method. We compare the performance from using the above different types of feature representation in each experiment, to study the effect of cluster feature representation on document metadata extraction, and to compare different word clustering methods in the tasks of document metadata extraction.

The header dataset[15] contains 935 labeled headers of computer science research papers, with 500 headers for training and 435 headers for testing. The headers are text files converted from the pdf and ps files. Each line ends with a carriage return and is marked for identification. The bibliographic dataset [15] contains 500 labeled references, and is randomized before being split into 250 training samples and 250 test samples. Each word of the header or a reference is labeled with a metatag as mentioned in Section 2.1.

## 4.2 Evaluation

We use two methods to evaluate the performance of document header line classification and bibliographic fields extraction. **Overall word classification accuracy** defines the percentage of the words that are tagged with their true label. **Class-specific evaluation** is achieved by Precision, Recall, Accuracy and F Measure. Let $A$ be the number of true positive samples predicted as positive, $B$ be the number of true positive samples predicted as negative, $C$ be the number of true negative samples predicted as positive, and $D$ be the number of true negative samples predicted as negative. A sample in this paper refers to a line in the line classification (Section 4.3), or a word in the bibliographic fields (Section 4.4). We define:

$$Precision = \frac{A}{A+C} \quad Recall = \frac{A}{A+B} \quad Accuracy = \frac{A+D}{A+B+C+D}$$

$$FMeasure = \frac{2Precision*Recall}{Precision+Recall}$$

## 4.3 Document header line classification

We compare 13 different sets of experiments based on different types of feature representation. The first 6 sets of experiments consist of two different sets each, where punctuation and stemming are treated differently. Experiment 1 removes all non letter characters except "-". Experiment 2 separates the ending punctuation mark of each word as an independent feature, and keeps all the non-letter characters in a word, such as [ ] ( ) / ' " : etc. Experiment 3 separates all the punctuation marks and non-letter characters except "-" as independent features. Experiments 4-6 correspond to Experiments 1-3, except that the words are stemmed and the case

information of each word is removed. For each of the feature representation in Experiments 1-6, we apply Baker et al.'s distributional word clustering algorithm and replace the original words by cluster labels. Experiments [1-6]W are conducted on the pre-processed word feature representation. Experiments [1-6]D are conducted on the feature representation after the distributed word clustering. We use four cluster sizes (100, 200, 500 and 600) in each of the Experiments [1-6]D and show the best line classification performance. Experiment 7 uses only our cluster feature representation.

Our cluster feature representation reduces the original 11223 word features to 588 cluster features. Since most of the cluster features are different combinations of Dig[N], Cap[M], etc., the number of the basic features is about 200. Therefore we choose the cluster size 100, 200, 500 and 600 in distributional word clustering for a better comparison between the two word clustering methods.

Besides the above word-specific features, we also add the statistical features for the clusters, i.e., the percentage of a certain cluster feature in the sample. For example, if a five-word line has two address words, we set a statistical feature *:Addrper:* with the value 0.4 (2/5). We have the following 14 statistical features for the cluster feature representations: *:DatePer:, :DictWordPer:, :NonDictWordPer:, :Cap1DictWordPer:, Cap1NonDictWordPer:, :DigitPer:, :AffiPer:, :AddrPer:, :PhonePer:, :DegreePer:, :PubnumPer:, :NotePer:, :CapPer:, :OthersPer:* For the distributional word cluster representation, we define a statistical feature for each cluster. If the cluster size is 100, we have 100 statistical features.

For space limitation, we only report the F Measure of the 13 experiments (Table 2). We observe from the result table that: (1) Our cluster-feature representation achieves the highest F Measure in most classes, especially classes of "pubnum", "address", "author", and "title". On average, our rule-based word clustering shows 6.6% absolute improvement over the original words representation, and 5.4% absolute improvement over the word distributional word clustering. (2) Author class and note class are where our method achieves highest and least performance increase compared to the original words feature representation. Table 3 shows the features ranked by expected entropy loss in two different types of feature representation for both classes. The first column shows that the top-ranked original words for author class are name initials and name words. The highest ranked cluster feature ":name word:" and the 4th ranked feature ":SingleCap:" shown in the second column summarize all the original word features shown in the first column. Cluster features such as :degree: and :Dig[4]:, are from negative training data and have high discriminatory power, as calculated by the expected entropy loss. These probably explain the performance increase our cluster method achieves. (3) Our cluster feature representation degrades the line classification for some cases, such as the "note" class. An explanation is that over-generalization of the words loses information and hurts the classification performance. (4) Taking punctuation marks and non-letter characters as separate features improves the classification performance, e.g. in the classes of "email" and "web".

## 4.4 Bibliographic field extraction

The rule-based feature representation achieves the overall bibliographic field extraction accuracy 89.9%, which has the 8.4% absolute improvement than using the original words representation. The rule-based feature representation reduces the feature dimensionality from 2300 original words to 300 clusters. We cluster the original bibliographic words into 300 clusters using distributional word clustering. However, the distributional cluster feature representation achieves lower accuracy (77.5%).

Table 4 shows that the cluster feature representation improves the overall class-specific extraction performance, especially for classes "editor", "page", "tech" and "volume". The hidden Markov model learned from the clustered training data seems to generalize better than when just using the original work representation. Calculating the emission probabilities of cluster-specific features rather than the original words increases the probability of emitting each word in the domain databases corresponding to the cluster-specific features, even the probability of the words not seen in the training data. E.g., a single sample containing only Author1 in the author field will also increase the probability of seeing his co-authors in that state.

| Author Class | | Note Class | |
|---|---|---|---|
| **Original words** | **Cluster features** | **Original words** | **Cluster features** |
| . | :name word: | by | :note: |
| A. | :Cap1NonDictWord: | supported | :1010: |
| David | :DictWord: | Research | :0110: |
| J. | :SingleCap: | part | :Dig[4]: |
| M. | :Cap1DictWord: | was | :CapWords:-:Digs: |
| P. | :note: | NSF | :Cap1DictWord: |
| S. | . | grant | :Mix[6]:-:Digs:-:CapWords:-:Digs: |
| R. | :NonDictWord: | under | . |
| E. | , | ACM | :Mix[6]:-:Digs:-:Digs:-:Digs: |
| for | : | Proceedings | :CapWords:-:CapWords: |
| K. | :degree: | research | :MixCaseWords: |
| H. | :Dig[4]: | Defense | :0011: |
| D. | :Dig[5]: | Agency | , |
| John | :CapWord[2]: | Conference | :DictWord: |
| Michael | :CapWord[3]: | contract | :Dig[2]:-:Dig[2]: |

**Table 3: Top ranked features (before and after wording clustering) in author class and note class, computed by the expected entropy loss. Original words and cluster features do not have 1-1 correspondence.**

| Bib field | Original words | | | Distributional clusters | | | Our clusters | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F | P | R | F | P | R | F |
| author | 96.3 | 87.0 | 91.5 | 87.2 | 97.6 | 92.1 | 96.2 | 99.1 | **97.6** |
| book title | 92.4 | 88.6 | **90.4** | 81.7 | 87.3 | 84.4 | 88.7 | 88.9 | 88.8 |
| date | 87.9 | 82.2 | 84.9 | 87.6 | 82.2 | 84.8 | 98.5 | 95.9 | **97.2** |
| editor | 76.8 | 45.2 | 56.9 | 68.5 | 60.7 | 64.4 | 81.7 | 63.7 | **71.6** |
| institution | 68.4 | 78.8 | 56.9 | 78.3 | 71.2 | 74.6 | 76.5 | 77.3 | **76.9** |
| journal | 89.3 | 65.2 | 75.4 | 61.0 | 63.1 | 62.0 | 77.1 | 78.7 | **77.9** |
| location | 76.5 | 75.5 | **76.0** | 78.8 | 71.5 | 75.0 | 77.7 | 71.5 | 74.5 |
| note | 58.1 | 57.4 | 57.8 | 32.7 | 39.4 | 35.7 | 76.3 | 47.9 | **58.8** |
| pages | 71.0 | 73.5 | 72.2 | 66.0 | 74.0 | 70.0 | 95.6 | 96.9 | **96.2** |
| publisher | 81.3 | 60.0 | 68.9 | 68.5 | 72.4 | **70.4** | 56.0 | 58.6 | 57.3 |
| tech | 12.2 | 79.5 | 21.1 | 15.9 | 97.4 | 27.4 | 56.2 | 64.1 | **59.9** |
| title | 87.9 | 84.0 | 85.9 | 96.0 | 61.8 | 75.2 | 92.2 | 93.0 | **92.6** |
| volume | 85.2 | 73.2 | 78.7 | 81.8 | 60.4 | 69.5 | 87.7 | 91.3 | **89.5** |

**Table 4: Bibliographic field words tagging performance(%) using different feature representations. P-Precision, R-Recall and F-F Measure.**

The effect of rule-based word clustering may be constrained by the structure of the models used. The HMM transition structure learned from training data is fixed, regardless of feature representation. State transitions depend only on the labels of the classes of the training data. If a transition from the class "author" to the class "title" is not present in the training samples, rule-based word clustering will not help predicting such transition for test samples. Taking single words as observations in a first-order HMM model may also restrict the exploitation of the domain databases, e.g., a single word is unable to match a multi-word country name.

# 5. CONCLUSION AND DISCUSSION

This paper introduces a new rule-based word clustering method for feature representation. The domain databases and word orthographic properties embed domain knowledge into the problem and help form the clusters.The new cluster feature representation shows significant performance gain and dimensionality reduction in document header line classification and bibliographic field extraction. Such cluster feature representation outperforms the representations that use original words and that use distributional word clustering in both experiments. Comparing to distributional word clustering, our method appears to have computational advantages since we only need to search the domain databases and check word orthographic properties using simple rules.

Word clustering is a way of generalizing words. It expresses the concepts underlying the word clusters. However, over-generalization loses the specific information of each word. For example, replacing

| Class | 1W | 1D | 2W | 2D | 3W | 3D | 4W | 4D | 5W | 5D | 6W | 6D | 7C | Increase | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Title | 77.9 | 81.3 | 77.8 | 81.9 | 78.1 | 81.8 | 80.8 | 86.0 | 79.2 | 82.5 | 82.9 | 84.8 | **92.2** | 11.4 | 6.2 |
| Author | 61.5 | 69.6 | 64.1 | 74.7 | 62.2 | 69.6 | 62.1 | 68.0 | 65.8 | 73.3 | 62.1 | 68.9 | **92.3** | 26.5 | 17.6 |
| Affiliation | 89.8 | 90.2 | 90.8 | 90.4 | 89.9 | 89.3 | 90.3 | 90.1 | 90.7 | 90.5 | 90.2 | 89.8 | **91.6** | 0.8 | 1.2 |
| Address | 81.1 | 80.2 | 82.4 | 81.0 | 82.3 | 80.7 | 78.1 | 79.5 | 82.6 | 81.0 | 82.1 | 80.3 | **92.3** | 9.9 | 11.3 |
| Note | 69.3 | 69.0 | 69.2 | 68.5 | 70.2 | 67.4 | 68.9 | 70.0 | 70.4 | 70.0 | 69.0 | **70.6** | 64.8 | -5.6 | -5.8 |
| Email | 92.0 | 51.9 | 27.5 | 30.8 | **98.7** | 95.3 | 92.0 | 88.4 | 27.9 | 31.4 | **98.7** | 97.7 | 98.1 | -0.6 | 0.4 |
| Date | 83.1 | 79.2 | 85.1 | 86.0 | 83.0 | 79.3 | 82.0 | 83.1 | 85.1 | 87.6 | 83.3 | 86.1 | **93.6** | 8.5 | 6.0 |
| Abstract | 96.8 | 97.2 | 96.8 | 97.1 | 97.0 | 97.2 | 95.5 | 95.7 | 95.7 | 95.6 | 95.6 | 95.5 | **97.5** | 0.5 | 0.3 |
| Phone | 63.5 | **78.9** | 63.5 | 64.6 | 65.6 | 77.8 | 76.1 | 75.0 | 75.4 | 76.1 | 77.1 | 78.9 | **78.9** | 1.8 | 0 |
| Keyword | 65.7 | 68.9 | 65.1 | 67.3 | 66.3 | 68.2 | 64.8 | 66.0 | 64.1 | 65.4 | 65.1 | 66.0 | **69.5** | 3.2 | 0.6 |
| Web | 96.0 | 94.1 | 36.4 | 36.4 | 96.0 | 96.0 | 96.0 | 90.6 | 35.3 | 25.8 | 96.0 | 96.0 | **96.2** | 0.2 | 0.2 |
| Degree | 55.6 | 60.9 | 57.1 | 62.3 | 57.1 | 60.9 | 58.5 | 57.3 | 61.1 | 58.1 | 60.1 | 58.0 | **64.2** | 3.1 | 1.9 |
| Pubnum | 52.6 | 50.0 | 53.3 | 53.3 | 52.6 | 50.0 | 53.3 | 50.6 | 55.3 | 52.0 | 53.3 | 50.6 | **81.6** | 26.3 | 29.6 |

**Table 2: Line classification performance of document headers based on different feature representations, evaluated by F Measure (%). "W" refers to the experiments on pre-processed original words; "D" refers to the experiments on the features after Baker et al.'s distributional word clustering; "C" refers to the experiment using our cluster feature representation. The best result is marked by bold font. The two sub columns of the "Increase"column show the performance gain our cluster feature representation (column 7C) has over the best performance achieved by original word representation (left column) and distributional word clustering (right column).**

all the digits by ":number:" may degrade the capability of distinguishing "month" and "pubnum". It would be an interesting research issue to study how to measure the degree of generalization.

The domain databases and proper use of word orthographic properties are important for effective word clustering. Inappropriate or small domain databases may introduce various biases in word clustering. The choice of domain databases and different word orthographic properties currently is context dependent and is done manually. It would be interesting to explore approaches that automatically select domain databases and find useful word orthographies.

## Acknowledgments

We gratefully acknowledge Andrew McCallum for providing the HMM code and Cheng Li for useful suggestions throughout the experiments. We would like to acknowledge partial support from NSF grant NSDL 0121679, CCF 0305879 and Lockheed Martin.

## 6. REFERENCES

[1] L. D. Baker and A. K. McCallum. Distributional clustering of words for text classification. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval*, pages 96–103, 1998.

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

[3] K. Dave, S. Lawrence, and D. M. Pennock. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International World Wide Web Conference*, 2003.

[4] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[5] I. Dhillon, S. Manella, and R. Kumar. A divisive information-theoretic feature clustering for text classification. *Machine Learning Research(JMLR)*, 2002.

[6] E. Glover, G. Flake, S. Lawrence, W. P. Birmingham, A. Kruger, C. L. Giles, and D. Pennock. Improving category specific web search by learning query modifications. In *Proceedings of the Symposium on Applications and the Internet, SAINT*, pages 23–31, 2001.

[7] H. Han, C. L. Giles, E. Manavoglu, H. Zha, Z. Zhang, and E. A. Fox. Automatic document metadata extraction using support vector machines. In *Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, 2003.

[8] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence*, 1999.

[9] D. Lin and P. Pantel. Document clustering with committees. In *Conference on Computational Linguistics*, pages 577–583, 2002.

[10] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

[11] T. Mitchell. Version spaces: A candidate elimination approach to rule learning. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pages 305–310, 1977.

[12] F. C. N. Pereira, N. Tishby, and L. Lee. Distributional clustering of english words. In *Meeting of the Association for Computational Linguistics*, pages 183–190, 1993.

[13] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 77(2):257–286, 1989.

[14] P. Schone and D. Jurafsky. Knowlege-free induction ofinflectional morphologies. In *Proceedings of the North American chapter of the Association for Computational Linguistics (NAACL-2001)*, 2001.

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

[16] N. Slonim and N. Tishby. The power of word clusters for text classication. In *Proceedings of the 23rd European Colloquium on Information Retrieval Research*, 2001.

[17] A. Takasu. Bibliographic attribute extraction from erroneous references based on a statistical model. In *Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, pages 49–60, 2003.

[18] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[19] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the 14th International Conference on Machine Learning*, pages 412–420, 1997.