Oriental Journal of Computer Science & Technology Vol. **1**(1), 15-24 (2008)

Improving on the smoothing technique for obtaining emission probabilities in hidden Markov models

BOLANLE A. OJOKOH*, OLUMIDE S. ADEWALE and SAMUEL O. FALAKI

Department of Computer Science, Federal University of Technology, P.M.B. 704, Akure (Nigeria)

(Received: February 12, 2008; Accepted: April 04, 2008)

ABSTRACT

Hidden Markov Models (HMMs) have been shown to achieve good performance when applied to information extraction tasks. This paper describes the training aspect of exploring HMMs for the task of metadata extraction from tagged bibliographic references. The main contribution of this work is the improvement of the technique proposed by earlier researchers for smoothing emission probabilities in order to avoid the occurrence of zero values. The results show the effectiveness of the proposed method.

Key words: Hidden markov model**s,** parameters, emission probabilities, Smoothing, Non-zero values

INTRODUCTION

Hidden markov modeling is a powerful statistical learning technique with widespread application in various areas (Rabiner and Juang, 1986). Hidden Markov models (HMMs) have been shown to achieve good performance when applied to information extraction tasks in both semi structured and free text. The main advantage of HMMs in language modeling is the fact that they are well suited for the modeling of sequential data, such as spoken or written language. Another serious motivation behind the use of HMMs for text-based tasks is their strong statistical foundations, which provide a sound theoretical basis for the constructed models. In addition, HMMs are computationally efficient to develop. They are the most widely used generative learning method for representing and extracting information from sequential data, hence have been applied with significant success to many text-related tasks, including part-of-speech tagging (Kupiec, 1992), text segmentation and event tracking (Yamron *et al.,* 1998), named entity recognition (Bikel *et al.,*1997) and information extraction (Leek, 1997; Freitag and McCallum, 1999 and 2000; Han *et al.*,, 2003) (Ojokoh *et al.,* 2007). Some researchers have applied HMMs to metadata extraction but no work has been discovered where HMM was applied for the specific purpose of evaluating HMM's performance for metadata extraction for the same set of tagged references used in this research. HMM is particularly suited for metadata extraction because of its state transition matrix which is very good at catching correlated features because it is capable of keeping the probabilities from one state to another state.

HMMs provide a natural framework for modeling the production of the bibliography of research papers. The goal of metadata extraction here using HMMs is to label each word of a reference as belonging to a class such as author, title, journal, volume, or date. The entire reference (and all the classes to extract) is modeled with one HMM. In the process of obtaining values for the model parameters, this paper presents an improvement over the smoothing technique earlier suggested. Taghva *et al.* (2005), in the process of applying HMMs to the task of address extraction used absolute discounting to smooth emission probabilities. They used the method proposed by Borkar *et al.,* (2001). It was also discovered that the smoothing technique (shrinkage) suggested by Freitag and MacCallum (1999) did not in any way improve HMM for address extraction; it rather brought degradation. After using the method proposed by these researchers for smoothing, it was discovered that zero values still exist for some of the emission probabilities, hence this research.

The rest of this paper is structured as follows. In section 2 we review the basic theory of HMMs and its application to the problem to be solved. In section 3, we present the implementation and the results obtained and especially the contribution of this research to the smoothing technique. Finally, we conclude in section 4, discussing further research efforts.

HMMS for metadata extraction

Basic theory of hidden markov models

The hidden markov model is a five-tuple (S, A, Π, V, B) (Rabiner, 1989),

H = (S, A, V, B) ...(1)

$$S = \{S_1, \dots, S_N\} \qquad \text{...(2)}$$

where,

N is the number of states

$A = \{a_{ij}\}$ is the probability of transitioning from state i to state j ...(3)

where,

$$\sum_{j=1}^{N} a_{ij} = 1,\ 1 \le i \le N \qquad \text{...(4)}$$

$\Pi = \{P[S_i(1)]\}$ indicating the probability of being at state S_i at time t=1 ...(5)

$$V = \{V_1, \dots, V_M\} \qquad \text{...(6)}$$

where M is the number of emission symbols in the discrete vocabulary, V.

$B = \{b_j(k)\}$ indicating the probability of observing symbol, V_k at state S_j, ...(7)

where,

$$\sum_{k=1}^{M} b_j(k) = 1,\ 1 \le j \le N \qquad \text{...(8)}$$

This emission probability matrix, B was obtained in this research using Maximum Likelihood as follows:

$$P(V|S)_{ml} = \frac{\text{Number of times symbol V is emitted at state S}}{\text{Total number of symbols emitted by state S}}$$

However, maximum likelihood estimation was supplemented with smoothing because the maximum likelihood estimation will sometimes assign a zero probability to unseen emission-state combinations. Absolute discounting was used to smooth emission probabilities by Borkar *et al.* (2001) and Taghva *et al.* (2005). Absolute discounting consists of subtracting a small amount of probability *p*, from all symbols assigned a non-zero probability at a state S. Probability *p* is then distributed equally over symbols given zero probability by the maximum likelihood estimate.

If v is the number of symbols assigned a non-zero probability at a state S and N is the total number of symbols, emission probabilities are calculated by:

$$P(V|S) = \begin{cases} P(V|S)_{ml} - p & \text{if } P(V|S)_{ml} > 0 \\ \dfrac{vp}{N-v} & \text{otherwise} \end{cases} \qquad \text{...(10)}$$

where,

$$p = \frac{1}{T_s + v}$$

and

T_s is the total number of symbols emitted at state S that is the denominator of $P(V|S)_{ml}$. This research proposes a modification of Borkar et al. and Taghva et al.' s method, discovering that using their method may still lead to some zero probabilities for some of the matrix elements, thus, the equations used for smoothing were:

$$P(V|S) = \begin{cases} P(V|S)_{ml} - p & \text{if } P(V|S)_{ml} > 0 \text{ and } P(V|S)_{ml} > p \\ P(V|S)_{ml} & \text{if } P(V|S)_{ml} > 0 \text{ and } P(V|S)_{ml} <= p \\ \dfrac{vp}{N-v} & \text{otherwise} \end{cases}$$

...(11)

where,

$$p = \frac{1}{T_s + v}$$

The problem to be solved further is to obtain the most possible state sequence, $Q = \{q_1, q_2, ..., q_T\}$ for a given observation sequence, $O = \{O_1, O_2, ..., O_T\}$. This can be obtained by computing the probabilities for all possible sequences and then obtaining the maximum probability as follows:

$$\delta_t(i) = \max_{q_1 \cdots q_{t-1}} P\left[q_1 q_2, ..., q_t = i, O_1, O_2, ..., O_t | \lambda\right] \quad ...(12)$$

HMMs for metadata extraction from tagged references

This research revolves around automatically determining the model structure from data sets but focuses on the task of information extraction from tagged bibliographic references from cora data sets (http://www.cs.umass.edu/~mccallum/data/cora-ie.tar.gz). HMMs provide a natural framework for modeling the production of the bibliography of research papers. The goal of metadata extraction here using HMM is to label each word of a reference as belonging to a class such as author, title, journal, volume, or date. The entire reference (and all the classes to extract) is modeled with one HMM.

A selected example of tagged reference from the data set is shown in Fig. 1.

Below the line are the hidden states, “author”, “title” ,”journal”, “volume”, and “date” (five out of the thirteen existing in the data sets). Above the line are observation symbols, “pname”, “comma”, “fullstop”, and “startcap” (four out of the twenty four identified in the data sets). Arrows are used to indicate the state transitions or the symbol emissions from a state. After learning from the

<author>W. Landi and B.G. Ryder**</author><title>**Aliasing with and without pointers: A problem taxonomy.**</title><institution>**Center for Computer Aids for Industrial Productivity**</institution><tech>**Technical Report CAIP-TR 125, **</tech> <institution>**Rutgers University,**</institution><date>**September 1990.**</date>**

Fig. 1: Example of tagged reference from data set

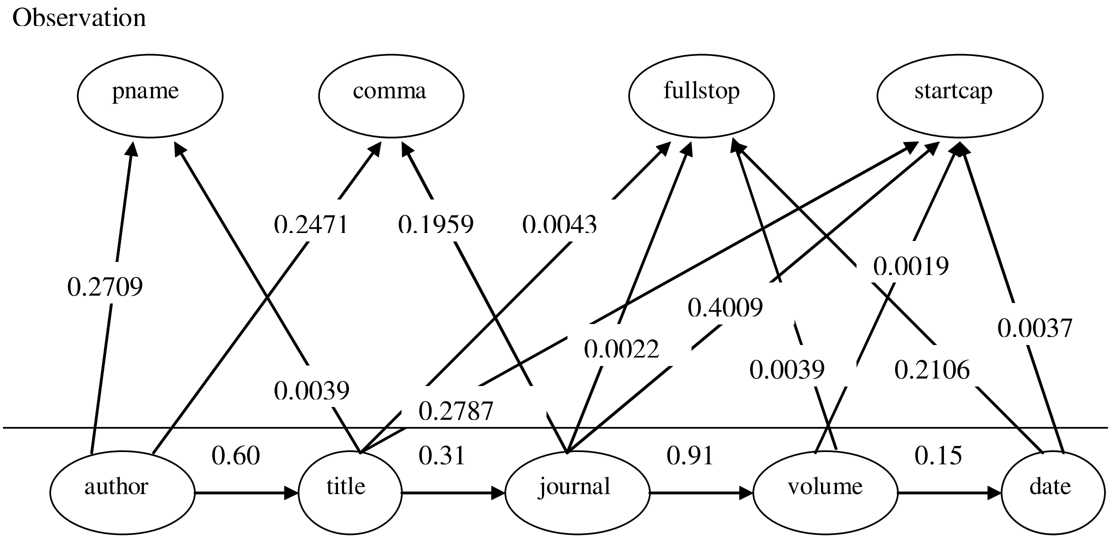

Fig. 2: Describes an example HMM from the data set

samples, the HMM estimated that the probability, for example, of a transition from “author” to “title” is 0.60. The probability of transition from “title” to “journal” is 0.31. In “author” state, the probability of observing “pname” is 0.2709. In “title” state, the probability of observing “fullstop” is 0.0043.

Table 1: Emission probability matrix values by maximum likelihood (for pname, comma, fullstop, initial, startcap, hyphen, containsnumber, and purenumber symbols)

	pname	comma	fullstop	initial	startcap	hyphen	contains number	pure number
author	0.2713	0.2475	0.0426	0.3168	0.0301	0.0004	0.0000	0.0000
title	0.0043	0.0331	0.0047	0.0000	0.2791	0.0000	0.0004	0.0026
editor	0.1469	0.2322	0.0190	0.1896	0.1801	0.0000	0.0000	0.0000
booktitle	0.0000	0.0994	0.0007	0.0000	0.4118	0.0000	0.0090	0.0404
date	0.0000	0.0531	0.2125	0.0000	0.0055	0.0000	0.0000	0.3516
journal	0.0000	0.1982	0.0000	0.0068	0.4032	0.0000	0.0000	0.0023
volume	0.0000	0.3711	0.0039	0.0000	0.0000	0.0000	0.0000	0.5039
tech	0.0067	0.2400	0.0000	0.0000	0.2333	0.0000	0.0400	0.1067
institution	0.0072	0.1986	0.0000	0.0000	0.3755	0.0000	0.0000	0.0000
pages	0.0000	0.1982	0.0061	0.0000	0.0030	0.0061	0.0000	0.5335
location	0.0031	0.3520	0.0031	0.0374	0.1464	0.0000	0.0000	0.0062
publisher	0.1081	0.2432	0.0000	0.0000	0.4932	0.0000	0.0000	0.0000
note	0.0000	0.0485	0.0194	0.0000	0.3301	0.0000	0.0194	0.0194

Table 2: emission probability matrix values by maximum likelihood (for statename, institutename, degreelike, quote, journalike, volumelike, datelike, and pagelike symbols)

	statename	institute name	degreelike	quote	journalike	volumelike	datelike	pagelike
author	0.0008	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0017
title	0.0013	0.0000	0.0004	0.0280	0.0000	0.0000	0.0000	0.0120
editor	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
booktitle	0.0000	0.0007	0.0015	0.0000	0.0000	0.0000	0.0000	0.0030
date	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1520	0.0000
journal	0.0000	0.0023	0.0000	0.0000	0.0991	0.0000	0.0000	0.0113
volume	0.0000	0.0000	0.0000	0.0000	0.0000	0.0938	0.0000	0.0000
tech	0.0000	0.0133	0.1200	0.0000	0.0000	0.0000	0.0000	0.0000
institution	0.0758	0.1841	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
pages	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.2530
location	0.3583	0.0031	0.0000	0.0000	0.0000	0.0031	0.0000	0.0000
publisher	0.0000	0.0068	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
note	0.0000	0.0000	0.0000	0.0000	0.0000	0.0097	0.0000	0.0388

A fully detailed algorithm of the HMM extraction process for the data sets is presented below:

Algorithm 1: Training.

´ Obtain the data sets to be used for the training of the extraction process

´ Use the training set to model the extraction

Table 3: Emission probability matrix values by maximum likelihood (for pardate, fullcolon, acronym, parenthesis, websitelike, ampersand, booklike, and otherwords symbols)

	pardate	fullcolon	acronym	parent hesis	website like	Amper-sand	booklike	otherwords
author	0.0000	0.0000	0.0050	0.0000	0.0000	0.0255	0.0000	0.0584
title	0.0000	0.0194	0.0374	0.0000	0.0004	0.0004	0.0004	0.5755
editor	0.0000	0.0047	0.0332	0.0379	0.0000	0.0047	0.0000	0.1517
booktitle	0.0000	0.0015	0.0448	0.0060	0.0000	0.0007	0.1368	0.2436
date	0.2033	0.0000	0.0037	0.0147	0.0000	0.0000	0.0000	0.0000
journal	0.0000	0.0045	0.0991	0.0000	0.0023	0.0000	0.0045	0.1667
volume	0.0039	0.0000	0.0039	0.0078	0.0000	0.0000	0.0000	0.0117
tech	0.0000	0.0000	0.0667	0.0000	0.0000	0.0000	0.1000	0.0733
institution	0.0000	0.0000	0.0108	0.0000	0.0000	0.0000	0.0000	0.1480
pages	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
location	0.0000	0.0467	0.0280	0.0000	0.0031	0.0000	0.0000	0.0093
publisher	0.0000	0.0000	0.1284	0.0000	0.0000	0.0068	0.0000	0.0135
note	0.0000	0.0097	0.0680	0.0000	0.0194	0.0000	0.0291	0.3883

Table 4: Emission probability matrix values after smoothing (for pname, comma, Fullstop, initial, startcap, hyphen, containsnumber, and purenumber symbols)

	pname	comma	fullstop	initial	startcap	hyphen	contains number	pure number
author	0.2709	0.2471	0.0422	0.3164	0.0296	0.0004	0.0004	0.0004
title	0.0039	0.0327	0.0043	0.0009	0.2787	0.0009	0.0004	0.0022
editor	0.1422	0.2275	0.0142	0.1848	0.1754	0.0032	0.0032	0.0032
booktitle	0.0010	0.987	0.0007	0.0010	0.4111	0.0010	0.0082	0.0396
date	0.0009	0.0513	0.2106	0.0009	0.0037	0.0009	0.0009	0.3498
journal	0.0022	0.1959	0.0022	0.0045	0.4009	0.0022	0.0022	0.0023
volume	0.0019	0.3672	0.0039	0.0019	0.0019	0.0019	0.0019	0.5000
tech	0.0067	0.2333	0.0045	0.0045	0.2267	0.0045	0.0333	0.1000
institution	0.0036	0.1949	0.0014	0.0014	0.3718	0.0014	0.0014	0.0014
pages	0.0010	0.1951	0.0030	0.0010	0.0030	0.0030	0.0010	0.5305
location	0.0031	0.3489	0.0031	0.0343	0.1433	0.0035	0.0035	0.0031
publisher	0.1014	0.2365	0.0027	0.0027	0.4865	0.0027	0.0027	0.0027
note	0.0087	0.388	0.0097	0.0087	0.3204	0.0087	0.0097	0.0097

process using One HMM.

´ Identify the number of states, N each state, S_i, being represented by the set of metadata to be extracted from the reference data sets.

´ Obtain the transition probabilities, a_{ij}, that is, the probability of transitioning from one state to another. The probability of transitioning from i to j is obtained by dividing the number

Table 5: Emission probability matrix values after smoothing (for statename, institutename, degreelike, quote, journalike, volumelike, datelike, and pagelike symbols)

	statename	institute name	degreelike	quote	journalike	volume like	datelike	page like
author	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0013
title	0.0009	0.0009	0.0004	0.0275	0.0009	0.0009	0.0009	0.1116
editor	0.0032	0.0032	0.0032	0.0032	0.0032	0.0032	0.0032	0.0032
booktitle	0.0010	0.0007	0.0007	0.0010	0.0010	0.0010	0.0010	0.0022
date	0.0009	0.0009	0.0009	0.0009	0.0009	0.0009	0.1502	0.0009
journal	0.0022	0.0023	0.0022	0.0022	0.0968	0.0022	0.0022	0.0090
volume	0.0019	0.0019	0.0019	0.0019	0.0019	0.0898	0.0019	0.0019
tech	0.0045	0.0067	0.1133	0.0045	0.0045	0.0045	0.0045	0.0045
institution	0.0722	0.1805	0.0014	0.0014	0.0014	0.0014	0.0014	0.0014
pages	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010	0.2500
location	0.3551	0.0031	0.0035	0.0035	0.0035	0.0031	0.0035	0.0035
publisher	0.0027	0.0068	0.0027	0.0027	0.0027	0.0027	0.0027	0.0027
note	0.0087	0.0087	0.0087	0.0087	0.0087	0.0097	0.0087	0.0291

Table 6: Emission probability matrix values after smoothing (for pardate, fullcolon, Acronym, parenthesis, websitelike, ampersand, booklike, and otherwords symbols)

	pardate	fullcolon	acronym	parent hesis	website like	Amper-sand	booklike	otherwords
author	0.0004	0.0004	0.0046	0.0004	0.0004	0.0250	0.0004	0.0580
title	0.0009	0.0189	0.0370	0.0009	0.0004	0.0004	0.0004	0.5751
editor	0.0032	0.0047	0.0284	0.0332	0.0032	0.0047	0.0032	0.1469
booktitle	0.0010	0.0007	0.0441	0.0052	0.0010	0.0007	0.1360	0.2429
date	0.2015	0.0009	0.0018	0.0128	0.0009	0.0009	0.0009	0.0009
journal	0.0022	0.0023	0.0968	0.0022	0.0023	0.0022	0.0023	0.1644
volume	0.0039	0.0019	0.0039	0.0039	0.0019	0.0019	0.0019	0.0078
tech	0.0045	0.0045	0.0600	0.0045	0.0045	0.0045	0.0933	0.0667
institution	0.0014	0.0014	0.0072	0.0014	0.0014	0.0014	0.0014	0.1444
pages	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010	0.0010
location	0.0035	0.0436	0.0249	0.0035	0.0031	0.0035	0.0035	0.0062
publisher	0.0027	0.0027	0.1216	0.0027	0.0027	0.0068	0.0027	0.0068
note	0.0087	0.0097	0.0583	0.0087	0.0097	0.0087	0.0194	0.3786

of transitions from i to j by the number of transitions from i. This is an NXN matrix computed manually from the samples.

- Obtain the initial probability matrix (1XN), that is the probability that state, S_i is the start state.
- Obtain the emission vocabulary from the sample.

Study the samples and identify some symbols that will fully represent every string (letter, word, numbers or characters) in the set of references. These form the emission vocabulary.

- Replace every string (except the tags) found in the sample by an emission symbol. This task will be very difficult done manually. Hence, the proposed algorithm to be implemented is described as follows:

```
For each reference[i] (where i = 1 to n) do
For each metadata[j] (where j = 1 to m) do
For each string[k] (where k = 1 to r) do
replace string[k] by an emission symbol      emission vocabulary by:
```

- replacing by a symbol directly (e.g for fullstop);
- replacing using regular expression (e.g for startcap);
- replacing by a symbol found from a database of objects; or a combination of two methods above

Obtain the emission probability matrix (by Maximum Likelihood estimation) each element being the probability of observing a symbol at a state as follows:

```
For each reference[i] (where i = 1 to n) do
For each metadata[j] (where j = 1 to m) do
For each symbol[k] (where k = 1 to r) do

count the number of times symbol[k] appears in metadata [j]
assign to count_symbol[k]
count the number of symbols[k] emitting from metadata[j]
assign to count_metadata [j]
prob[j,k] = count_symbol[k]/count_metadata [j]
```

Hence, a j x k matrix is obtained, forming the probabilities for all possible sequences.

In order to remove the occurrence of zero probabilities which may affect the result, performing smoothing on the result obtained (as proposed in this research) and obtain a new j x k matrix to be used for the testing process as follows:

```
(obtain values for Ts[j] and count_nonzero[j])
count_nonzero[j] = 0
    For each metadata[j] (where j = 1 to m) do
    For each symbol[k] (where k = 1 to r) do
    Ts[j] = count_metadata[j]
    If  prob[j,k] > 0 then
    count_nonzero[j] = count_nonzero[j] + 1
    (obtain the new jxk matrix)
    For each metadata[j] (where j = 1 to m) do
    For each symbol[k] (where k = 1 to r) do
    p[j] = 1/(Ts[j]+ count_nonzero[j])
    If  prob[j,k] > 0 and prob[j,k] > p[j] then
    new_prob[j,k] = prob[j,k] – p[j]
    elseif  prob[j,k] > 0 and prob[j,k] <= p[j] then
    new_prob[j,k] = prob[j,k]
    else    new_prob[j,k]= (count_nonzero[j] *p[j])/(r- count_nonzero[j])
```

Use this result to extract metadata from ∈the testing reference sets as follows:

Algorithm 2: Testing.

- Obtain the data sets to be used for the testing the model.
- Get the values for the model parameters obtained from the training process.
- Replace every string found in the sample by an emission symbol and remove the tags.
- For every reference (a sequence of words), obtain the best possible sequence of transitions and extract the words corresponding to every state.
- Perform an evaluation of the metadata extraction process.

RESULTS

Implementation

The work on HMMs centered on the task of extracting metadata from the references of some computer science research papers. A reference appears in a list of works at the end of a document, and provides full bibliographic information about a cited work (Powley and Dale, 2007). The reference of a research paper consists of author, title, editor, date and pages among others. Automatically extracting fields such as these for example is useful in constructing a searchable database of computer science research papers.

The model was trained, tested and evaluated using the cora data set (http://www.cs.umass.edu/~mccallum/data/cora-ie.tar.gz) consisting of 500 tagged references with 17,775 word tokens. This data set was used because of its public availability and relatively widespread usage. The first three hundred references consisting of 10,755 word tokens were selected for training while the rest two hundred consisting of 7,020 word tokens were used for testing. This was done arbitrarily with no special algorithm used (no explicit partitioning scheme has been published for comparison issues by other researchers (Krämer et al. (2007)) for grouping.

$$A = \begin{bmatrix}
0 & 0.60 & 0 & 0.01 & 0.39 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.04 & 0.42 & 0 & 0.31 & 0 & 0.14 & 0.01 & 0 & 0.02 & 0.06 & 0 \\
0 & 0.2 & 0 & 0.52 & 0.08 & 0 & 0.08 & 0 & 0 & 0 & 0.04 & 0.08 & 0 \\
0 & 0 & 0.01 & 0 & 0.16 & 0 & 0.16 & 0 & 0.01 & 0.41 & 0.2 & 0.07 & 0 \\
0 & 0.78 & 0.01 & 0.01 & 0 & 0.01 & 0 & 0.01 & 0.01 & 0.14 & 0 & 0 & 0.04 \\
0 & 0 & 0 & 0 & 0.02 & 0 & 0.91 & 0 & 0 & 0.03 & 0 & 0 & 0.03 \\
0 & 0 & 0.01 & 0.04 & 0.15 & 0 & 0 & 0 & 0 & 0.80 & 0 & 0.01 & 0 \\
0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0.78 & 0.03 & 0.06 & 0 & 0.03 \\
0 & 0 & 0 & 0 & 0.57 & 0 & 0 & 0.03 & 0 & 0.03 & 0.27 & 0 & 0.1 \\
0 & 0 & 0.01 & 0 & 0.63 & 0 & 0 & 0 & 0.01 & 0 & 0.28 & 0.04 & 0.02 \\
0 & 0 & 0 & 0.01 & 0.64 & 0 & 0 & 0 & 0.03 & 0.01 & 0 & 0.27 & 0.03 \\
0 & 0 & 0.03 & 0 & 0.33 & 0 & 0 & 0 & 0 & 0.17 & 0.47 & 0 & 0 \\
0.25 & 0 & 0.25 & 0 & 0.38 & 0 & 0 & 0 & 0.13 & 0 & 0 & 0 & 0
\end{bmatrix}$$

In the training process, values were obtained for the parameters of the HMM from the data sets using the earlier outlined algorithm as follows:

- N= 13, as there are 13 metadata (fields) needed to be extracted. These include author, title, editor, booktitle, date, journal, volume, tech, institution, pages, location, publisher and note. Therefore, S= $\{S_1,\ldots S_N\}$ where S_1 = author, S_2 = title,   = editor and so on.
- The transition probability matrix, $A = a_{ij}$, where   is the probability of transitioning from state i to j, where $1 \le i \le N$ and $1 \le j \le N$ was obtained as follows. For example,

The probability of transitioning from author to title

$$= \frac{\text{the number of transitions from author to title}}{\text{the number of transitions from author}}$$

This was computed from the sample manually. After all, a 13X13 matrix was obtained as follows:

$$\Pi = [0977 \quad 0 \quad 0.02 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0.003 \quad 0 \quad 0 \quad 0 \quad 0]$$

...(15)

The initial probabiity matrix, Π is a 1X13 it obtained as follow

´ The emission vocabulary, V, consists of 24 vocabulary as follows:
V= {pname, comma, fullstop, initial, startcap, hyphen, containsnumber, purenumber, statename, institutename, degreelike, quote, journallike, volumelike, datelike, pagelike, pardate, fullcolon, acronym, parentheses, websitelike, ampersand, booklike, otherwords}

After the taxonimising process. The emission probability matrix, B_{old} (a 13X24 matrix) was estimated initially by Maximum Likelihood. The values of the matrix are shown in Tables 1-3.

After this, smoothing was done to remove the cases of zero probabilities using the smoothing technique recommended by Borkar and Taghva after improvement (by this research).

Then, the emission probability matrix, B_{new} used for testing was obtained. The matrix values are shown in Tables 4-6.

It can be seen from the results in the tables that the existence of zero values have been removed from the emission probabilities as Tables 4 – 6 show.

CONCLUSION

In this research, HMM was particularly implemented for a sub-task of information extraction (metadata extraction from tagged bibliographic references). The training data set consists of 300 tagged references. From these, values were obtained for the HMM parameters.

To obtain the emission probability matrix without zero values, smoothing of the values was done. An improvement was made on the results obtained by earlier researchers. With the new results, better performance is expected of the hidden markov model for the task of metadata extraction. In the future, testing will be done and the system will be evaluated. In addition, the need to incorporate final state probabilities into the Hidden markov model could be considered in the future.

REFERENCES

1. Bikel, D., Miller, S., and Weischedel, R., Nymble: a high-performance learning name-finder. In Proceedings of ANLP-97, 194-201 (1997).
2. Borkar, V., Deshmukh, K., and Sarawagi, S., Automatic segmentation of text into structured records. In ACM SIGMOD 2001, (Santa Barbara, California, USA) (2001).
3. Freitag, D., and McCallum., Information extraction with HMMs and shrinkage. In Proceedings AAAI-99 Workshop Machine Learning and Information Extraction (1999).
4. Freitag D., McCallum A.K., Information extraction with HMM structures Learned by Stochastic Optimization, AAAI-2000, 584-589 (2000).
5. Han, H., Giles, C.L., Manavoglu, E., Zha, H., Zhang, Z., and Fox, E.A., Automatic Document Metadata Extraction Using Support Vector Machine. Joint Conference on Digital Libraries (JCDL'03), Houston, Texas USA (2003).
6. http://www.cs.umass.edu/~mccallum/data/cora-ie.tar.gz . Cora data set.
7. Kramer, M., Kaprykowsky, H., Keysers, D., Breuel, T., Bibliographic Meta-Data Extraction Using Probabilistic Finite State Transducers (2007).
8. Kupiec, J., Robust part-of-speech tagging using a hidden Markov model. Computer Speech and Language, **6**: 225–242 (1992).
9. Leek, T., Information extraction using hidden

markov models. Master's thesis, UC San Diego (1997).

10. Ojokoh, B.A., Falaki, S.O., and Adewale, O.S. Automated Information Extraction System for Heterogeneous Digital Library Documents. Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL 2007) Doctoral Consortium, Vancouver, British Columbia, Canada, June, 18-23 (2007).
11. Powley, B. and Dale, R., Evidence-Based Information Extraction for High Accuracy Citation and Author Name Identification Conference RIAO2007, Pittsburgh PA, U.S.A. May 30-June 1, 2007 (2007).
12. Rabiner, L., and Juang B. (1986). An introduction to hidden Markov models. IEEE Acoustics, Speech & Signal Processing Magazine, 3 ,4-16.
13. Rabiner, L. R., A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE, **77**(2): 257–286 (1989).
14. Taghva, K., Coombs, J., Pereda, R., and Nartker, T., Address Extraction using Hidden Markov Models. Information Science Research Institute, USA (2005).
15. Yamron, J., Carp, I., Gillick, L., Lowe, S., and van Mulbregt, P., A hidden Markov model approach to text segmentation and event tracking. In Proceedings of the IEEE ICASSP Seattle, Washington (1998).