2011 International Conference on Computer Communication and Management
Proc .of CSIT vol.5 (2011) © (2011) IACSIT Press, Singapore

Address Standardization using Supervised Machine Learning

ABDUL KALEEM [1], KHAWAJA MOYEEZULLAH GHORI [2] +, ZAHRA KHANZADA [3], M. NOMAN MALIK [4]
[1,2,3,4] National University of Modern Languages, Islamabad, Pakistan

ABSTRACT.
Data mining has become an important task of today's rich information environments. Strong results are obtained through accurate historical reporting. Inaccurate and dirty records yield weak and wrong analysis. Unfortunately, organizations store addresses in unstructured formats resulting in multiple representations of same entities. These addresses need to be cleansed before they can be used in mining data. In this paper we present a supervised machine learning procedure, Hidden Markov Model (HMM). This automated probabilistic approach is used to segment a set of Asian addresses into their atomic units and standardize them. Results of this technique show that it can also be used to standardize even a large set of complex and un-formatted addresses.
KEYWORDS
Hidden Markov Model (HMM), Viterbi Algorithm, address standardization, Machine learning.

▪ 1. Introduction

Structuring and atomization of data is decisive management fad as far as the organization operational efficiency and effectiveness is concerned. Management decisions about the data are very crucial and time oriented as new hidden patterns will help in decisions [1]. Information mining from soiled data is overwhelming chore in today's business ambiance. The presentation of data means the data must be in unique format in data warehouse, regarding storage of the data and data mining for further queries and interpretation. [1][2][10]

Data warehousing is a promising field as well as prevailing amplifier as it renovate little trivia into huge erroneous information that needs to be rinse out before it enters into a warehouse. The dirty data creates haphazard situation as it contain wrong, inconsistent, missing and different naming convention values which lead to misapprehend and unstructured information [1][2][3].

Data which never get any treatment and having original syntax contain many ambiguities and errors which cannot be processed for the storage and further execution. Quality of data is essential part of any data warehouse for extracting and storing data. A researcher illustrates it with the name of horrific quality of data as it endures with errors and irregularity [4].

This paper presents a probabilistic model called Hidden Markov Model (HMM) [2] to segment a set of Asian addresses into its atomic units. This machine learning procedure also uses Viterbi algorithm [3] to find the most probable path for a testing address. The success results for house holding are discussed in section six.

▪ 2. Related work

For the purpose of warehouse construction, the addresses have got to be clean and transformed to a standard format. This requires a process followed by multiple steps. First is Address Elementization [5] where addresses placed as a standard and structured format. Author [3] describes how an address is broken into its atomic units like house number, street number, city, state and zip etc. The second step is address standardization where format and mistakes get corrected and after that the addresses are grouped according to some criteria.

+ Corresponding author. Tel.: + 92-333-5185049;
E-mail address: moizghauri@yahoo.com

The last decade is very valuable as far as the data cleaning and their formatting is concerned as many theories and models emerged into the market. Authors [7] specify addresses as key part of any organizational data which must be kept in proper manner as mining data [8] [9] is the crucial phenomena.

3. Proposed Model

The input to the model is a collection of unstructured addresses of Pakistan Telecommunication Corporation Limited (PTCL) Telephone directory with various atomic units like house, street, sector, city, building name etc. Figure 1 describes our model.

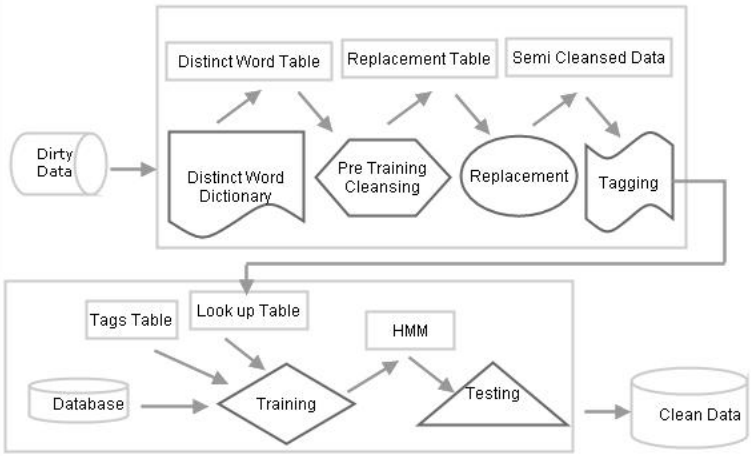

Fig. 1: The Model.

It consists of two main modules:

3.1. PRE TRAINING CLEANSING AND TAGGING

This module is used to identify multiple spellings of words, non-standardize abbreviations and replaces them by correct words. It consist of following four components

Distinct word dictionary creates a table “Dictionary” in the database. This table contains all the distinct tokens of address field in the original table. An address is tokenized on the basis of char, digits, delimiters and strings [2]. This table is used to get the names of shops, streets, organizations, and buildings etc. Pre training cleansing identifies different variations of words, typographical errors, misspelling and non-standardized abbreviations. A standard word is assigned to all variations. It also removes the meaningless words from the database. Sorted neighborhood method [11] is used to identify multiple variations of words. Following is the description of this approach:

A key for each word in the list is computed by extracting relevant portion of the word. An example of a key generation is to ignore all vowels and consecutive repetition of consonants as described in Table 1.The words in the database are then sorted by using the key found in the first step. A fixed size window is moved through the sequential list of records in order to limit the comparisons for matching records to those records in the window. If the size of the window is w records then every new record that enters that window is compared with the previous w -1 records to find matching record. The first record in the window slides out of it.

Word	Key
RAWALPINDI	RWLPND
COLONY	CLNY
RAWLPINDI	RWLPND
MUHMMAD	MHMD
MOHMMAD	MHMD
RAWALPIND	RWLPND
CLONY	CLNY
CLNY	CLNY
MOHMAD	MHMD

Table 1: Creation of keys

Word	Key
RAWALPIND	RWLPND
RAWLPINDI	RWLPND
RAWALPINDI	RWLPND
COLONY	CLNY
CLONY	CLNY
CLNY	CLNY
MUHMMAD	MHMD
MOHMMAD	MHMD
MOHMAD	MHMD

Table2: Sorted data

Table 2 lists records sorted and grouped on the basis of key generated

In replacement component, all words with typographical errors, incorrect spellings and non-standardized abbreviations are replaced in the database by correct words. Semi cleansed data is produced as a result. All distinct words of semi cleansed data are then assigned tags in tagging component on the basis of the state in which they are observed. We used four basic data types (digit, character, string, and delimiter) as symbol in Symbol emission probability matrix called Matrix B [3]. Table 3 lists some possible tags for Pakistani addresses.

Name	Description	Possible Values
HS	String for house	House, HNO, Bungalow
SS	String for street	STNO, Streetno, street
CHS	Character for house	A, B, H, C
CSE	Character for sector	H, F, G, I,
D1	Digit of length one	1, 2, 3
D2	Digit of length two	22,68
CM	Comma	,
CO	Colon	:
SL	Slash	/

Table 3: Example set of tags for Pakistani addresses.

3.2. MACHINE LEARNING

Once semi cleansed data are assigned tags, it undergoes into two phases namely training and testing. Training includes a probabilistic approach, Hidden Markov Model (HMM) [2][3] which is used to train the system using a set of randomly selected records. A probabilistic finite automaton is constructed after training is completed. Hidden Markov Model as discussed in [2][3] for record elementization is based on three probability matrices.

In testing phase, rest of the data is automatically segmented into its relevant states on the basis of training. Viterbi algorithm [3] is used to find the best path for each testing record. It is a dynamic programming algorithm that picks the best probability out of many paths, given an observation sequence.

▪ 4. Modified viterbi algorithm

Viterbi algorithm initializes Viterbi Matrix by product of ∏[i] and Bi,0 , Where ∏[i] is the initial probability matrix for the state i and Bi,0 is the emission probability of symbol 0 at state i. Viterbi assumes that both initial probability matrix and symbol emission probability matrix must have non zero values. Since our system starts testing from the second address so most of the states and symbols have zero probability so Viterbi Matrix is initialized with zero values. To overcome this problem we modified the initialization step of Viterbi algorithm. Now, if any of the two operands is a zero value, multiply the other operand with 0.0001 and if both are zero then Viterbi is initialized with constant value 0.0001 as shown below

$V_{0,i} = B_{0,i} \,.\, \prod_i \qquad if\ B_{0,i} != 0\ ,\ \prod_i != 0$

$V_{0,i} = B_{0,i} \,.\, 0.0001 \quad if\ B_{0,i} != 0\ ,\ \prod_i = 0$

$V_{0,i} = 0.0001 \cdot \prod_i$ $\quad if\ B_{0,i} = 0\ ,\ \prod_i\ != 0$

$V_{0,i} = 0.0001$ $\quad if\ B_{0,i} = 0\ ,\ \prod_i\ = 0$

▪ 5. Main features

Testing in our approach is some what unusual. It doesn't start after training is completed; it starts after training the first record. The second record is tested on the probabilities of first record. If it is segmented correctly its probabilities are added and number of trained records is incremented. Otherwise correct segmentation is manually applied. So training and testing goes side by side to minimize manual effort.

Most of the states were having values with similar data types, so the role of matrix B (discussed in [3]) was negligible and matrix A (discussed in [3]) started dominating. To avoid this concept, separate tags for each state were used. Each state had its own strings and characters but common digit and delimiter.

We tested our model on somewhat more complicated data set of Pakistani addresses as compared to well structured European addresses and got interesting results.

▪ 6. Experimental Results

We tested our system on two data sets. The results were as follows:

A database consisting of 15378 addresses of Islamabad was given as input to the system as address set 1, to check working of our system on large set of addresses. 615 records were trained. The rest 14763 were tested on the probabilities of training set. 30 states were identified. 83% percent records were segmented correctly.

A database consisting of 438 addresses of Islamabad was given as input to the system as address set 2, to check working of our system on smaller set of addresses. 6 records were trained. The rest 432 records were test on the probabilities of training set. 5 states were identified. The success ratio was 95%.

The size of training set is crucial in evaluating the results. We trained 2%, 4% and 5% of 15378 addresses and got success of 78%, 83%, and 83% respectively. It shows that HMM is a fast learner but after achieving certain peek performance it slows down. Similarly in case of address set 2, 1.4% and 3% of training produced almost same results.

The nature of data plays a vital role in evaluating the result. The data with lesser states needs low percentage of training to produce results with greater accuracy. For example in Address set 1, and with 5% training, 30 states were identified for addresses. We achieved 83% of success. Whereas in address set, with 1.4% training, 5 states were identified and we achieved 95% of success. So this difference in success is due to the number of states. The behavior of addresses varies as number of states gets larger as shown in Table 4.

Data Type	Total records	Training	Tested	States	Success
Address	15378	2%	98%	30	83%
Address	15378	4%	96%	30	83%
Address	15378	5%	95%	30	83%
Address	438	1.4%	98.6%	5	95%
Address	438	3%	97%	5	96%

Table 4: Results

Figure 2 shows the graph of house holding for 15378 addresses. The number of household groups is much more in cleansed data as compared to dirty data or semi-cleansed data.

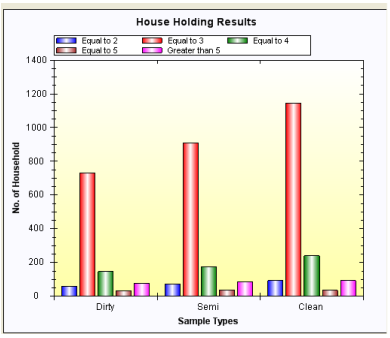

Fig.2: House holding

References

[1] Mong Li Lee, Hongjun Lu, Tok Wang Ling and Yee Teng Ko. Cleansing Data for Mining and Warehousing. Lecture Notes in Computer Science, 1999, Volume 1677/1999, 807, DOI: 10.1007/3-540-48309-8_70
[2] Vinayak R. Borkar, Kaustubh Deshmukh, Sunita Sarawagi. Automatically Extracting Structure from Free Text Addresses. Bulletin of the Technical Committee on Data Engineering December 2000 Vol. 23 No. 4
[3] Vinayak R. Borkar, Kaustubh Deshmukh, Sunita Sarawagi. Automatic segmentation of text into structured records, Proceedings of the 2001 ACM SIGMOD international conference on Management of data.
[4] Erhard Rahm, Hong Hai Do. Data Cleaning: Problems and Current Approaches Bulletin of the Technical Committee on Data Engineering, December 2000 Vol. 23 No. 4
[5] Ralph Kimball, Dealing with dirty data. Journal DBMS Volume 9 Issue 10, Sept. 1996.
[6] http://datamining.anu.edu.au
[7] Peter Christen , Daniel Belacic. Automated probabilistic address standardisation and verification. Australasian Data Mining Conference', AusDM'05, 2005.
[8] Han, J. and Kamber,, M. : "Data Mining: Concepts and Techniques", 2006.
[9] Rahm, E. and Do, H.H.: Data Cleaning: Problems and Current Approaches. IEEE Data Engineering Bulletin, 2000.
[10] Vicenç Torra, Information Fusion - Methods and Aggregation Operators, Part 6, 999-1008, DOI: 10.1007/978-0-387-09823-4_52, 2010
[11] Mong L. Lee, Tok W. Ling, Hongjun Lu, Yee Ko. Cleansing Data for Mining and Warehousing. Proceedings of the 10th International Conference on Database and Expert Systems Applications 1999, pp. 751-760.