MARKOV MODEL FOR DISCOVERING KNOWLEDGE IN TEXT DOCUMENTS

[1]I.BERIN JEBA JINGLE, [2] Dr. J.JEYA A.CELIN

[1]Assistant Professor, Department of Computer Science and Engineering, NIU

[2]Professor, Department of Information Technology, NIU

E-mail: [1]berinjeba@gmail.com, [2]jjeyacelin@gmail.com

ABSTRACT

The digital data knowledge discovery and data mining due to their immense growth have engrossed a great deal of deliberation in recent years. Numerous applications, such as market investigation and business society, can be promoted by use of the information and facts extracted from a bulky amount of data. Text mining is the skill that allows users to mine useful information from a large amount of digital text documents on the Web or databases.Text Mining is the discovery of unknown information, by automatically extracting information from different written resources. This paper deals with the patterns discovery in text documents using the hidden Markov model. In most existing pattern mining in text mining methods, all go through the problems of lack of accuracy and lack of performance. In the proposed system the first progression is pre-processing step which removes the “noise” word. The next development is the HMMs, which is used for pattern extraction and classification of input data. Hidden Markov Model calculates the possibility value between noticed events and unnoticed events. This method can improve the accuracy of evaluating term weights and also used to progress the performance for discovering patterns in text for large databases.

Keywords: *Data mining, HMM, Classification, stemming, Smoothing Technique*

1. INTRODUCTION

Text Mining is the process of automatically analysing text to extract information that is concern to a particular user or useful for a particular purpose. Text mining looks for patterns in text while data mining looks for patterns in data. It represents a new perspective to the common problem of finding relevant information. Example reasons for using text mining include: creating links between objects that mention the same event such as a person’s name, extracting metadata for a modern digital library, exploring how a market is evolving, and looking for more ideas or relations.

Text mining is about the use of statistical and machine learning techniques to learn structural elements of text in order to search for useful information in previously unseen text. It is an extension of data mining, which finds information in structured database, to the far less structured domain of free text. Using text mining tools, people are able to explore items which consist of one or more words, such as a person’s name and a name of location, in a large collection of documents without having to look through a great number of files, and to understand the given text in order to extract useful information from it.

Many text mining methods and algorithms are used for pattern discovery in text mining from large databases. One of the technique used is pattern based approach (i.e.) PTM [1] (Pattern Taxonomy methodology).The PTM has two process the pattern evolving and the pattern deploying, this overcomes the low frequency and the misinterpretation problem. But the discovered pattern faced the problem of lack of performance and lack of accuracy.

The next approach used for discovering patterns in text mining is the PTM and the Naïve Bayes Classifier [2]. This methods solves the low frequency and misinterpretation problems but, the complexity of this system increases hence the speed of operation goes down here. So the discovery of patterns experiences lack of performance problem here.

The next methodology used for discovering patterns in text mining is a Novel pattern mining approach [5]. It is used for mining frequent patterns in text mining. This method first mines patterns from negative and positive feedbacks and then classifies to find specific patterns and also to remove noise data. The next process is, it applies a Novel pattern deploying strategy which improves the performance of frequent patterns in text data. This approach is a time consuming process hence speed is decreased.

In the case of pattern discovery in text mining a new approach is developed, which first discovers closed sequential patterns in text data to discover the important informative substance of the documents. Next with the help of this identified information it mines useful patterns in text documents. Here a novel fusion method is also developed which is based on Dempster-Shafer's evidential reasoning to combine the documents to extract patterns. This model is able to process a high volume of documents and it does not require complex training process and parameter tuning. But this model lack in accuracy and Performance.

2. HIDDEN MARKOV MODEL:

A hidden Markov model is finite-state automaton with stochastic state transitions and symbol emissions. It is a particular model based on a sequence of events, and consists of a set of states and a set of output symbols. The automaton generates a sequence of symbols by starting from the initial state, transitioning to a new state, emitting an output symbol, transitioning to another state, emitting another symbol, and so on, until the final state is reached. HMMs are used for pattern extraction and primary classification of the given input data.

2.1 Data flow Diagram for the proposed methodology

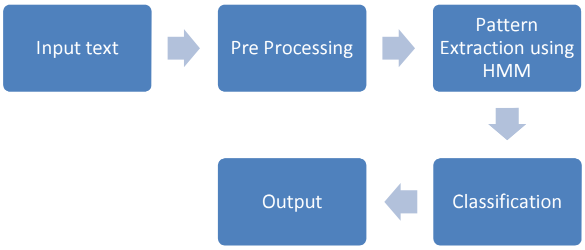


Figure 1: Proposed work data flow diagram

2.2 Input Data:

Application scenario used is Extraction of Addresses from text document. The input text file which contains addresses is used as the input data.\

2.3 Pre-Processing:

A large number of texts must be prepared, some of which are used for training the system, while others are used for evaluation purpose. Each text should be assigned to the relevant categories beforehand. The words obtained from the categorization form the basis for the feature space of the training data. Pre-processing not only prunes down the training size, makes the data more clean and capable of training the classifier more effectively. The conversion of the entire text to lower-case and removal of non-alphanumeric contents are done by Stop-word elimination and grammatical stemming. Many text classification systems use stop-word list to remove "noise" words before going to the classification algorithm.

The next step is to go in for grammatical stemming. A stemmer is an algorithm which identifies a stem form of a given word. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even. The purpose of stemming is to map different forms of an identically implying word to the same feature in the classifier‘s training space.

Since the knowledge base plays an important role in this type of classification system, the method of training set selection, generation and refinement of knowledge base are crucial problems. The knowledge base is build using Hidden Markov Model.

2.4 The HMM Definition

The Hidden Markov Model (HMM) is a variant of a finite state machine having a set of hidden states, Q, Hidden states Q = { qi }, i = 1, . , N an output alphabet (observations), O, Observations (symbols) O = { ok }, k = 1, . . . , M .

- Transition probabilities, A,
- output (emission) probabilities, B, and
- Initial state probabilities, π

The current state is not observable. Instead, each state produces an output with a certain probability (B). Usually the states, Q, and outputs, O, are understood, so an HMM is said to be a triple, (A, B, π). For each member of the set of states, {q1, q2… qN} there are two probability distributions. One governs the outgoing state transitions, which indicates how likely another state is to follow; the other governs the emission of symbols in the observation vocabulary O= {O1, O2,……,OM} which indicates how likely a symbol is to be generated in the particular state N and are the number of states and number of symbols respectively. We assume that time is discrete, and the model transitions between states at each time unit. The probability of moving from state Si to state Sj is stored in the state transition matrix, where: A = {aij} where aij = P (qj at t +1 | qi at t)}, where P(a | b) is the conditional probability of a given b, t = 1, . . . , T is time, and qi in Q. Informally, A is the probability that the next state is qj given that the current state is qi. When the HMM moves between states, it emits an output symbol after each transition. Exactly which output symbol is emitted depends on the output symbol distribution B, which defines the probability of emitting a particular symbol in a particular state. Emission probabilities B = {bik = bi (ok) = P (ok | qi)}, where ok in O. Informally, B is the probability that the output is ok given that the current state is qi. To complete the model we need an initial probability distribution π = {pi = P (qi at t = 1)}.

2.5 Applying HMM to Pattern Extraction:

For Pattern Extraction, the observation sequence is a sequence of words in text. The symbols emitted in each state are words, and the HMM is a word- level model. Each sequence corresponds to a sentence in text, and each state corresponds to a type of token that the program will identify and mark up. Each type of token will be marked in the text by a unique tag Xi. N, the number of states in the model, is the number of different pattern classes, and is determined by the training data. Because the system uses a word-level HMM model, M, the size of the output vocabulary, is the number of different words that appear in the training data. The matrix A, A gives the probability that the current word belongs to a particular pattern type given that the previous word belongs to a particular pattern type as well. Distribution B, B is the probability of the same words being seen in a particular pattern class. It is pattern-dependent: different pattern classes have different probabilities for a certain word. The initial distribution π is the probability that each type of pattern starts a sentence. The pattern identification task is as follows: given a sequence of words, identify the appropriate patterns and mark them up with predefined labels, given that a model has already been constructed. It is assumed that each individual word belongs to a class. Words and pattern class refer to observation symbols and states in the HMM. Finding tokens in a sequence of words means finding the state sequence that underlies the input. This is just the second of the three basic problems of HMM. The solution is discussed in the following section.

2.6 Estimate the parameters of the HMM

Once the model structure is determined, the next problem is to estimate the model parameters for the state transition probabilities A and state-specific lexical distribution B given a set of training data. Generally, there are two kinds of methods: unsupervised and supervised. For unsupervised learning, the training data is untagged—no labels are inserted into it. A and B can be estimated by applying the Baum-Welch algorithm. Given the initial parameters, this algorithm adjusts model parameters iteratively to maximize the likelihood of untagged data. However, because there can be different possible results, the corpus must be correctly analyzed before being used for parameter estimation. It requires great effort to analyze a large corpus manually. The result is also sensitive to the initial parameters, because the maximization is local. Supervised learning uses tagged training data that is, sequences of words with the target words already marked up with associated labels. The information required to construct the model can be obtained by recognizing the labels. But to obtain such a corpus requires a large amount of effort to tag tokens manually. While both methods involve some manual effort, analyzing the corpus may require more expertise than tagging. On the other hand, training data can be tagged by applying an automatic tagger to the raw material and checking the result manually. Also, the unsupervised method creates more complex models than the supervised one. Previous work has shown that supervised methods have been applied quite successfully to the task.

In the undertaken research, labeled training corpora are used; thus learning is supervised. Transitions from suffix states to any of the prefix states was allowed, and a final target state could transition to a prefix state if addresses appeared close together in training data. The original state transition probability and symbol emission distribution are calculated in a straightforward manner by using the ratio of counts, events/sample size or words/vocabulary:

Transition probabilities were estimated by Maximum Likelihood:

$$P(Si, Sj) = \frac{\text{Number of transition from Si to Sj}}{\text{Total Number of transisions out of Si}}$$

The emission probability table is estimated with Maximum Likelihood supplemented by smoothing. Smoothing is required because Maximum Likelihood estimation will sometimes assign a zero probability to unseen emission-state combinations. Prior to smoothing, emission probabilities are estimated by:

$$P\left(\frac{w}{s}\right)ml = \frac{\text{Number of times w emitted by state s}}{\text{total number of symbols emitted by s}}$$

2.7 Training the HMM

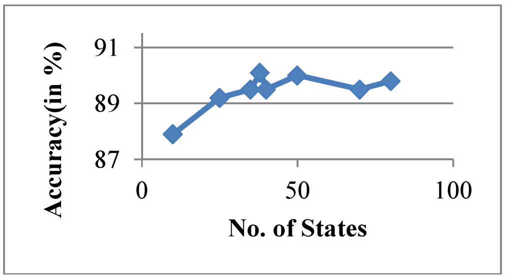
Before constructing the model, the training data is split into sentences. The system processes the training data in two passes. The first pass counts the number of token classes N, the number of different words M and the vocabularies for each pattern type. The second pass counts the number of events and calculates the A and B matrices.

2.8 Smoothing the Probabilities:

Smoothing is a technique for adjusting probability estimates that have been obtained from the training data. Smoothing is necessary when data sparse. It is especially important for handling the zero frequency problems, which is ever-present in models constructed by learning Zero frequencies occur in both contextual and lexical probabilities. Therefore the probability that the model transitions between the corresponding states is zero. The zero frequency problems occur more often in lexical estimation because of the nature of text: most words are plain text. The probabilities that these words are generated in other classes are zero

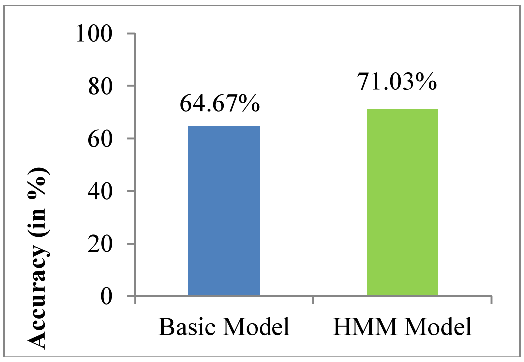
3. EXPERIMENTAL RESULT:

Extraction accuracy for Hmm based on number of states. This graph denotes the increased accuracy when compared with the existing PTM model. The PTM solves two problems low frequency and the misinterpretation problem. But the accuracy was slow hence with the HMM the accuracy is improved.



4. COMPARISON OF HMM WITH BASIC MODEL:

This histogram gives the comparison of the existing PTM model with the HMM model. The accuracy of the pattern discovery has increased than the existing PTM model.



5. CONCLUSION:

This paper presents the HMM model for extracting the patterns from large databases. The proposed work uses various modules like pre-processing, Pattern extraction, and classification for discovering patterns. The Hidden Markov model is a method which calculates the Probability values between the noticed and unnoticed events. The proposed work reported high accuracy when compared with the existing methods

REFERENCES:

[1] Ning Zhong, Yuefeng Li, and Sheng-Tang Wu, "Effective Pattern Discovery for Text Mining" IEEE Transactions vol. 24, no. 1, January 2012.

[2] Kavitha Murugeshan, Neeraj RK," Discovering Patterns to Produce Effective Output through Text Mining Using Naïve Bayesian Algorithm" International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-2, Issue-6, May 2013.

[3] Mrs.K.Mythili, Mrs K.Yasodha," A Pattern Taxonomy Model with New Pattern Discovery Model for Text Mining" International Journal of Science and Applied Information Technology Volume 1, No.3, ISSN No. 2278-3083July – August 2012.

[4] Charushila Kadu, Praveen Bhanodia, Pritesh Jain, "Hybrid Approach to Improve Pattern Discovery in Text mining"International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 6, June 2013.

[5] Luepol Pipanmaekaporn and Yuefeng Li," A Pattern Discovery Model for Effective Text Mining" Springer, 2012, pp-540

[6] Zhen Hai, Kuiyu Chang, Jung-Jae Kim, and Christopher C. Yang," Identifying Features in Opinion Mining via Intrinsic and Extrinsic Domain Relevance", IEEE Transactions on knowledge and data engineering, vol. 6, no. 6, June 2012.

[7],Miss Dipti S.Charjan, Prof. Mukesh A.Pund ," Pattern Discovery For Text Mining Using Pattern Taxonomy". International Journal of Engineering Trends and Technology (IJETT) – Volume 4 Issue 10- October 2013.

[8] Jian Ma, Wei Xu, Yong-hong Sun, Efraim Turban, Shouyang Wang, and Ou Liu, "An Ontology-Based Text-Mining Method to Cluster Proposals for Research Project Selection" International Journal of Engineering Trends and Technology (IJETT) – Volume 5 Issue 10- October 2011.

[9] Spyros I. Zoumpoulis, Michail Vlachos, Nikolaos M. Freris, Claudio Lucchese, "Right-Protected Data Publishing with Provable Distance-based Mining " IEEE Transactions on knowledge and data engineering, vol. 21, no. 19, november 2012.

[10] K. Aas and L. Eikvil, "Text Categorisation: A Survey," Technical Report Raport NR 941, Norwegian Computing Center, 1999.

[11] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," Proc. 20th Int'l Conf. Very Large Data Bases (VLDB '94), 1994, pp. 478-499.

[12] H. Ahonen, O. Heinonen, M. Klemettinen, and A.I. Verkamo, "Applying Data Mining Techniques for Descriptive Phrase Extraction in Digital Document Collections," Proc. IEEE Int'l Forum on Research and Technology Advances in Digital Libraries (ADL '98), 1998, pp. 2-11.

[15] N. Cancedda, E. Gaussier, C. Goutte, and J.-M. Renders, "Word- Sequence Kernels," J. Machine Learning Research, vol. 3, 2003, pp. 1059-1082.

[16] M.F. Caropreso, S. Matwin, and F. Sebastiani, "Statistical Phrases in Automated Text Categorization," Technical Report IEI-B4-07-2000, Instituto di Elaborazione dell' Informazione, 2000.

[17] C. Cortes and V. Vapnik, "Support-Vector Networks," Machine Learning, vol. 20, no. 3, 1995, pp. 273-297.

[18] S.T. Dumais, "Improving the Retrieval of Information from External Sources," Behavior Research Methods, Instruments, and Computers, vol. 23, no. 2, 1991, pp. 229-236.

[19] J. Han and K.C.-C. Chang, "Data Mining for Web Intelligence,"Computer, vol. 35, no.11, Nov. 2002, pp. 64-70.

[20] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation," Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '00), 2000, pp. 1-12.