STATISTICAL MARKOVIAN DATA MODELING FOR NATURAL LANGUAGE PROCESSING

Fawaz S. Al-Anzi and DiaAbuZeina

Department of Computer Engineering, Kuwait University, Kuwait City, Kuwait

ABSTRACT

Markov chain theory is a popular statistical tool in applied probability that is quite useful in modelling real-world computing applications. Over the past years; there has been grown interest to employ Markov chain theory in statistical learning of temporal (i.e. time series) data. A wide range of applications found to utilize Markov concepts; such applications include computational linguists, image processing, communications, bioinformatics, finance systems, etc .In fact, Markov processes based research applied with great success in many of the most efficient natural language processing (NLP) tools. Hence, this paper explores the Markov chain theory and its extension hidden Markov models (HMM) in (NLP) applications. This paper also presents some aspects related to Markov chains and HMM such as creating transition and observation matrices, calculating data sequence probabilities, extracting the hidden states, and profile HMM.

KEYWORDS

Markov chains, hidden Markov models, profile hidden Markov Models, natural language processing

1. INTRODUCTION

Markov chains theory is increasingly being adopted in real-world computing applications since it provides a convenient way for modelling temporal, time-series data. At each clock tick, the system moves into a new state that can be the same as the previous one. A Markov chain model is a statistical tool that captures the patterns dependencies in pattern recognition systems. For this reason, Markov chain theory is appropriate in natural langue processing (NLP) where it naturally characterized by dependencies between patterns such as characters or words. Reference [1] demonstrated the limitations of using hand-written rules in NLP applications, and the reasons why to move toward statistical approaches.

Markov chains are directed graphs (a graphical model) that generally used with relatively long data sequences for data-mining tasks. Such tasks include prediction, classification, clustering, pattern discovery, software testing, multimedia analysis, networks, etc. Reference [2] indicated that there are two reasons of Markov chains popularity; very rich in mathematical structure and work well in practice for several important applications. Hidden Markov models (HMM) are an extension of Markov chains that used to find the hidden system's states based on the observations. Consequently, the conventional HMM described as follows. Given a sequence of observations, based on the trained model that fit the training data best, find the hidden states that most likely have generated the observations.

In order to facilitate the research in this direction, this paper provides a survey of this so popular data modelling technique. However, because of the wide range of the research domains that uses this technique. We specifically focus on the NLP related applications. Reference [3] lists some

DOI: 10.5121/ijdkp.2017.7103

domains that utilize Markov chains theory which include: physics, chemistry, testing, speech recognition, information sciences, queuing theory, internet applications, statistics, economics and finance, social sciences, mathematical biology, genetics, games, music, baseball, text generators, bioinformatics. Reference [4] lists the five greatest applications of Markov chains that include Scherr's application to computer performance evaluation, Brin and Page's application to Page Rank and Web Search, Baum's application to HMM, Shannon's application to information theory, and Markov's application to Eugeny Onegin.

This paper organized as follows. The next section presents a background of Markov chains theory. Section 3 highlights the main concepts of HMM followed by description of profile HMM in section 4. In section 5, we present the literature review of both Markov chains and HMM. Finally, we conclude in section 5.

2. MARKOV CHAINS

In the early of twentieth century, Andrei Markov used his name to indicate for the theory he proposed, [5].Markov chains are quite popular in computational linguistics for data modelling. A Markov chain is a memory less stochastic model that describes the behaviour of an integer-valued random process. The behaviour is the simple form of dependency in which the next state (or event) depends only on the current state. According to [6], a random process said to be Markov if the future of the process, given the present, is independent of the past. To describe the transitions between states, a transition diagram used to describe the model and the probabilities of going from one state to another. For example, Figure 1 shows a Markov chain diagram with three states (Easy, Ok, and Hard) that belong to exam cases (i.e. states). In the figure, each arc represents the probability value for transition from one state to another.

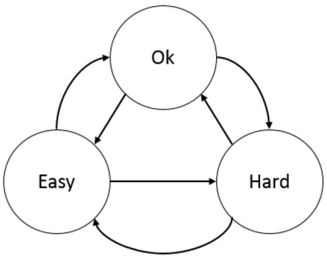

Figure 1. A Simple Markov chain with three states

Markov chain diagrams are generally represented using state transition matrices that denote the transition probabilities from one state to another. Hence, a state transition matrix is created using the entire states in the system. For example, if a particular textual application has a training data that contains N states (e.g. the size of lexicon), then the state transition matrix is described by a matrix A= {aij} of size N*N. In matrix A, the element aij denote the transition probability from a state i to a state j. Table 1 shows how the state transition matrix used to characterize the Markov diagram shown in Figure 1. That is, the matrix carries the state transitions probabilities between the involved states (Easy, Ok, and Hard). For illustration, the P(E|H) denote to the probability of the next exam to be Easy given that the previous exam was Hard.

Table 1. A state transition matrix of three states

State		Next Exam		
		Easy (E)	Ok (O)	Hard (H)
Previous Exam	Easy (E)	P(E|E)	P(O|E)	P(H|E)
	Ok (O)	P(E|O)	P(O|O)	P(H|O)
	Hard (H)	P(E|H)	P(O|H)	P(H|H)

In Table 1, the sum of the probability values at each row is 1 as the sum of the probabilities coming out of each node should be 1. Hence, P(E|E)+P(O|E)+P(P(H|E) equal 1. Markov chain is a worthy topic that has many details. For examples, it contains discrete-time, continuous-time, time-reversed, reversible, and irreducible Markov chains. The case shown in Figure 1 is irreducible case, also called ergodic, where it is possible to go from every state to every state.
To illustrate a simple Markov chain data model, a small data set contains two English sentences used to create a transition matrix based on the neighbouring characters sequences. The sentences are inspirational English quotes picked from [7]:

(1) Power perceived is power achieved. (2) If you come to a fork in the road, take it.

Figure 2 shows the transition matrix of these two quotes by counting the total number of occurrences of the adjacent two character sequences. It is a 19 × 19 matrix where the number19 is the total number of unique characters appeared in the sentences (i.e the upper mentioned quotes). In this example, creating a transition matrix is case insensitive where D is same as d, as an example. In addition, a space between two words discarded and not considered in the transition matrix. Figure 2 shows that the maximum number in the matrix's entries is 3 (a highlighted underlined value) which means that moving from character e to r (e➔r) is the most frequently appeared sequence in this small corpus. The words that contain this sequence are :{ Power (two times) and perceived}.

	a	c	d	e	f	h	i	k	m	n	o	p	r	s	t	u	v	w	y
a	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
c	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	0	0	2	0	0	0	1	0	0	0	0	0	**3**	0	0	0	1	0	0
f	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
h	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
i	0	0	0	1	1	0	0	0	0	1	0	0	0	1	1	0	1	0	0
k	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
m	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
o	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	2	0
p	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
r	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
u	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
w	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
y	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Figure 2. A transition matrix of two characters sequences

Based on the information obtained in the transition matrix shown in Figure 2. It is possible to answer some questions related to the given data collection. Among questions, what is the total unique number of the two characters sequences appeared in the given data set? What are the two characters sequences that did not appear in the data collection? Whatare the least frequently two characters sequences in the data set? Accordingly, Markov chains used as prediction systems such as weather forecasting. Therefore, it is possible to predict the tomorrow's weather according to the today's weather. For example, if we have two states (Sunny, Rainy), and the requirement is

to find the probability P(Sunny|Rainy), Markov chains make it possible based on the information provided in the probability transition matrix. Another example of the using Markov chains is banking industry. A big portfolio of banks based on loans. Therefore, Markov chains used to classify loans to different states such as Good, Risky, and Bad loans.

For simplicity, the information presented in Figure 2 shows the transition matrix based on total number of occurrences. Figure 3 shows the same information but using probabilities instead of the number of occurrences. That is, it contains the probability of moving from one character to another. As previously indicated, the sum of entries at each row is equal 1. In Figure 3, any matrix entry that has 0 means that there is no transition at that case. Similarly, if the matrix entry is 1, it means that there is only one possible output of that state. For example, the character “o” comes after “y”, and this is the only possible arc of the state “y”.

	a	c	d	e	f	h	i	k	m	n	o	p	r	s	t	u	v	w	y
a	0	0.33	0.33	0	0	0	0	0.33	0	0	0	0	0	0	0	0	0	0	0
c	0	0	0	0.33	0	0.33	0	0	0	0	0.33	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	0	0	0.29	0	0	0	0.14	0	0	0	0	0	0.43	0	0	0	0.14	0	0
f	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
h	0	0	0	0.5	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0
i	0	0	0	0.17	0.17	0	0	0	0	0.17	0	0	0	0.17	0.17	0	0.17	0	0
k	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
m	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
o	0.17	0	0	0	0	0	0	0	0.17	0	0	0	0.17	0	0	0.17	0	0.33	0
p	0	0	0	0.33	0	0	0	0	0	0	0.67	0	0	0	0	0	0	0	0
r	0	0.33	0	0	0	0	0	0.33	0	0	0.33	0	0	0	0	0	0	0	0
s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t	0.33	0	0	0	0	0.33	0	0	0	0	0.33	0	0	0	0	0	0	0	0
u	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
w	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
y	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Figure 3. A probability transition matrix of two characters sequences

3. HIDDEN MARKOV MODELS

HMMis an extension to Markov chains models as both used fortemporal datamodeling.The theory of HMM introduced by Baum and his colleagues in 1960s [8]. Reference [9] indicated that most current learning research in NLP employs particular statistical techniques inspired by research in speech recognition, such as HMM and probabilistic context-free grammars (PCFGs).However, the difference is that we observe the outputs in Markov chain, but the system states are hidden in HMM. Of course, the numbers of states and the observations have to be fixed and known. In this section, we explain the concept of HMM based on example provided in Figure 1 that shows a three exam’s states Markov diagram. As a simple example, supposed that a student’s parents want to know the levels (i.e the difficulty) of their son’s exams, naturally, it is possible to recognize the exam as Easy or Ok if the son feels Fine. Similarly, it is possible to recognize the exam as Hard if the son looks Scared. From the parents’ point of view, the required states (i.e. Easy, Ok, or Hard) are hidden. However, they directly observe the student’s reaction or feeling. Hence, the parents might use the observed reactions an indication to know the hidden states. HMM is described using three matrices: the initial probability matrix, the observation probability matrix, and the state transition matrix. Figure 4 shows a HMM diagram that shows the states and the observations. In the figure, each arc represents the probability between the states and between the states and the observations.

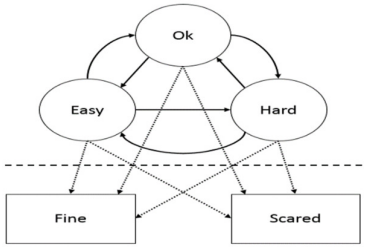

Figure 4. A HMM diagram with states and observations

Based on the information provided in the matrices, one can use either forward (also called any path) or Viterbi (also called best path) algorithms to find the probability scores during recognition phase. Figure 5 shows the trellis diagram for exam states HMM. While Forward algorithm used to compute the recognition probability of a sequence, Viterbi used to find the best-state sequence associated with the given observations, this process is also known as back-tracking. Hence, after computing the observations sequence probability and finding the maximum probability (supposed the star in Figure 5), the Viterbi algorithm leads the process back to identify the states (sources) from which the observations sequence have been emitted. In Figure 5, the maximum probabilities supposed to occur at the states shown using the dotted lines as follows; (starting at t=1): Hard, Easy, and Ok. Hence, the parents might consider the exams were Hard, Easy, and Ok, respectively.

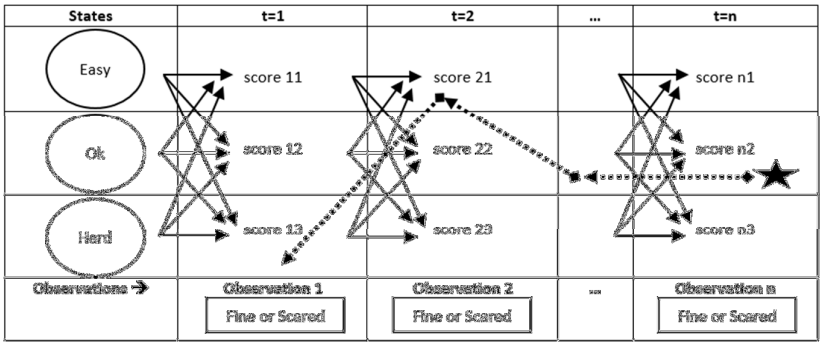

Figure 5.Trellis diagram of three states HMM

To illustrate how HMM employed to extract hidden states, Viterbi algorithm used to fine the probability at each time t. Reference [10] and other many references describe Viterbi algorithm. The example is regarding exam's states and the reactions of the student as explained in the previous sections. Figure 6 shows the HMM parameters and a question regarding the exam's states. As shown in the figure, at t=5, the values are small and they will continue decreasing as the observations increase that might lead to underflow problem. Reference [11] proposed solutions to the floating-point underflow problem that appear in the context of extremely small probability values when applying the Viterbi or forward algorithm to along sequences. The solution is to log all probability values and then add values instead of multiply for Viterbi algorithm. Regarding, Forward algorithm, the solution is to use scaling coefficients that keep the probability values in the dynamic range of the machine.

<table>
<tr><th>HMM parameters</th><th>A question and the answer</th></tr>
<tr><td>The transition matrix:
<table>
<tr><td></td><td colspan="3">next state</td></tr>
<tr><td></td><td>Easy</td><td>Ok</td><td>Hard</td></tr>
<tr><td>Easy</td><td>0.1</td><td>0.3</td><td>0.6</td></tr>
<tr><td>Ok</td><td>0.3</td><td>0.3</td><td>0.4</td></tr>
<tr><td>Hard</td><td>0.7</td><td>0.2</td><td>0.1</td></tr>
</table>
The observation matrix:
<table>
<tr><td></td><td>Easy</td><td>Ok</td><td>Hard</td></tr>
<tr><td>Fine</td><td>0.8</td><td>0.5</td><td>0.1</td></tr>
<tr><td>Scared</td><td>0.2</td><td>0.5</td><td>0.9</td></tr>
</table>
Theinitial matrix:
Π=[1/3,1/3,1/3]’</td>
<td>Given the following observation {Scared, Find, Fine, Scared, and Fine}; what are the most likely exam’s states. The following table shows the information obtained when implementing Viterbi algorithm.
<table>
<tr><td>States</td><td>α1(t=1)</td><td>α2(t=2)</td><td>α3(t=3)</td><td>α4(t=4)</td><td>α5(t=5)</td></tr>
<tr><td>Easy</td><td>0.0667</td><td>0.1680</td><td>0.0134</td><td>0.0015</td><td>0.0051</td></tr>
<tr><td>Ok</td><td>0.1667</td><td>0.0300</td><td>0.0252</td><td>0.0038</td><td>0.0009</td></tr>
<tr><td>Hard</td><td>0.3000</td><td>0.0067</td><td>0.0101</td><td>0.0091</td><td>0.0002</td></tr>
<tr><td>Observation</td><td>Scared</td><td>Fine</td><td>Fine</td><td>Scared</td><td>Fine</td></tr>
</table>
Where α is a variable that describe the probability at each time t.The shaded numbers indicated the max probability at time t. Accordingly, the exam’s states are :{ Hard, Easy, Ok, Hard, Easy}.Examples of some calculations: α1 (t=1, Scared) = 1/3*0.2 =0.0667. α1 (t=2, Fine) = max (0.0667*0.1, 0.1667*0.3, 0.3*0.7)*0.8 = max (0.006, 0.05, 0.21) * 0.8=0.168.</td></tr>
</table>

Figure 6.An example ofHMM and Viterbi calculations

3. PROFILE HIDDEN MARKOV MODELS

Even though HMM has been successfully used in linguistics such as speech recognition, however, it currently being used in modeling molecular biology sequences (e.g. genes and proteins) through what is called profile HMM. A profile HMM is a certain type of HMM that allowsposition dependent gap penalties. Hence, a profile HMM generally used for protein classification by creating a profile for each family through a sequence alignment process. The motivation of profile HMM is that it treats protein-spelling complexities in a systematic way. Figure 7show a profile HMM. As shown, profile HMM is a special type of Left-Right HMM (i.e. one direction) contains three stats: match, insert, and delete. In classification systems, the Baum-Welch (Forward-Backward) algorithms used for training and the Forward (any-path) algorithm used for scoring. Viterbi algorithm also used in profile HMM training and classification. Hence, profile HMM used to build an individual profile for each family and then find the max probability (i.e. the most likely family) of a molecular sequence, in question, given the model.

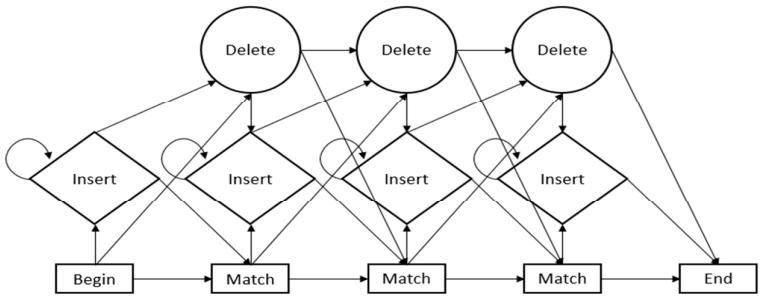

Figure 7. A profile HMM

Figure 8 shows examples of match insert, delete cases of some of molecular sequences. The match case in Figure 8 (a) shows the most conserved states (shaded area➔ACTAGT). After selecting the main states, the insert states specified as shown in Figure 8 (b). Hence, the match state characterized by high frequently observed symbols while insert states are the little observed one. Case (c) shows the delete states. As showing Figure 8 (c), the third sequence has a gap in the column 4 whereas this location has previously considered as a main states. Therefore, the case represented as a delete case. Similarly, the fourth sequence has a delete case.

(a) Match cases	(b) Insert cases	(c) Delete cases
AC-TA-GT	AC-TA-GT	AC-TA-GT
AC-TACGT	AC-TACGT	AC-TACGT
ACG-A-GT	ACG-A-GT	ACG-A-GT
A--TACGT	A--TACGT	A--TACGT
ACGTA-GT	ACGTA-GT	ACGTA-GT

Figure 8.Match, Insert, and Delete cases

4. LINGUISTIC APPLICATIONS

In the literature, there are quite many studies on modelling content dependencies for linguistics applications. Markov chain models and HMM are of great interest to linguistic scholars who primarily work on data sequences. Even though this study focuses on linguistic applications, however, Markov chains used to model a variety of phenomena in different fields. Figure 9 showssome of applications employed Markov chains. We intentionally ignored the references as the literature hastoo many studies employed Markov chains:

image processing, text and image compression, video segmentation ,forecasting, networking, signal processing, communications, software testing, genetics, bioinformatics, genome structure recognition, anomaly detection, tumour classification, water quality, epidemic spread, wind power, malicious and cyber-attack detection, traffic management, physics, chemistry, mathematical biology, games, music, multimedia processing, business activities, frauds detection.

Figure 9.Some of Markov chains and HMM applications

The following two subsections include some of the linguistic studies that utilized Markov chain theory. Linguistic applications topics mainly include (but not limited) speech recognition, speech emotion recognition, part-of-speech tagging, machine translation, text classification, text summarization, optical character recognition (OCR), named entity recognition, question answering, authorship attribution, etc. For the reader who interested in NLP, Reference [12] is a good reference as it demonstrates a thorough study of NLP (Almost) from Scratch.

4.1. NLP Markov Chains Based Research

The literature has a large number of studies that employ Markov chains for NLP applications. The following are some linguistic related applications. Reference [13] proposed a word-dividing algorithm based on statistical language models and Markov chain theory for Chinese speech processing. Reference [14] presented a semantic indexing Markov chains algorithm that uses both audio and visual information for event detection in soccer programs. Reference [15] investigated the use of Markov Chains and sequence kernels for the task of authorship attribution. Reference [16] implemented a probabilistic framework for support vector machine (SVM) that allows for automatic tuning of the penalty coefficient parameters and the kernel parameters via Markov chain for web searching via text categorization. Reference [17] demonstrated an automatic video annotation using multimodal Dirichlet process mixture model by collecting samples from the corresponding Markov chain. Reference [18] used a linguistic steganography detection method based on Markov chain models. Reference [19] showed how probabilistic Markov chain models used to detect topical structure in large text corpora.

Reference [20] proposed a method of recognizing location names from Chinese texts based on Max-Margin Markov network. Reference [21] utilized Markov chain and statistical language models in a linguistic steganography detection algorithm. Reference [22] proposed a Markov chain based algorithm for Chinese word segmentation. Reference [23] presented two new textual feature selection methods based on Markov chains rank aggregation techniques. Reference [24] proposed a Markov chain model for radical descriptors in Arabic text mining. Reference [25] presented statistical Markov chain models for the distributions of words in text lines. Reference [26] proposed a method for handwritten Chinese/Japanese text (character string) recognition based on semi-Markov conditional random fields (semi-CRFs). Reference [27] presented a Markov chain method to find authorship attribution on relational data between function words. Reference [28] utilized a probabilistic Markov chain model to infer the location of Twitter users. Reference [29] proposed a Markov chain based technique to determine the number of clusters of a corpus of short-text documents. Reference [30] proposed a Markov chain based method for digital document authentication. Reference [31] used Markov chain for authorship attribution in Arabic poetry. Reference [32] investigated the application of mixed-memory Markov models (MMMs) to automatic language identification. MMMs used to approximate standard statistical n-gram models ($n > 2$) by a mixture of bigram models.

4.2. NLP Hidden Markov Models Based Research

HMM based research has been for long an active research area due to the rapid development in NLP applications. The literature has many studies as follows. Reference [33] proposed to extract acronyms and their meaning from unstructured text as a stochastic process using HMM. Reference [34] proposed a morphological segmentation method with HMM method for Mongolian. Reference [35] employed HMM for Arabic handwritten word recognition based on HMM. Reference [36] presented a scheme for off-line recognition of large-set handwritten characters in the framework of the first-order HMM. Reference [37] proposed the use of hybrid HMM/Artificial Neural Network (ANN) models for recognizing unconstrained offline handwritten texts. Reference [38] used HMM for recognizing Farsi handwritten words. Reference [39] describes recent advances in HMM based OCR for machine-printed Arabic documents. Reference [40] proposed a HMM based method for named entity recognition. Reference [41] combined text classification and HMM techniques for structuring randomized clinical trial abstracts. Reference [42] employed HMM for medical text classification. Reference [43] proposes text (sequences of pages) categorization architecture based on HMM. Reference [44] described a model for machine translation based on first-order HMM. Reference [45] introduced speech emotion recognition by use of HMM. Reference [46] presented a HMM based method for speech emotion recognition. Reference [47] discussed the role of HMM in speech recognition. Reference [48] indicated that almost all present day large vocabulary continuous speech recognition (LVCSR) systems based on HMM. Reference [49] presented a text summarization method based on HMM. Reference [50] presented a method for summarizing speech documents using HMM. Reference [51] used HMM for part-of-speech tagging task. Reference [52] presented a second-order approximation of HMM for part-of-speech tagging task.

4.3. Profile Markov Models Based Research

Up to the date of writing this paper, no profile HMM based research found to serve NLP. Most of the works related to molecular applications.

5. CONCLUSIONS

In this paper, we presented Markov chains and HMM as standard models in language modelling. In the last decades, utilizing Markov and Hidden Markov based concepts have been steadily increasing in linguistic applications such as speech recognition, part of speech tagging, and noun-phrase chunking. This work discussed the potential and the size of Markov and Hidden Markov based research particularly related to NLP applications. For future work, it is worthy to compare the capabilities of HMM with other machine learning tools such as deep neural networks in building automatic speech recognition (ASR) systems.

ACKNOWLEDGEMENTS

This work is supported by Kuwait University Research Administration Research Project Number EO06/12.

REFERENCES

[1] Nadkarni, Prakash M., LucilaOhno-Machado, and Wendy W. Chapman. "Natural language processing: an introduction." Journal of the American Medical Informatics Association 18.5 (2011): 544-551.
[2] Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.
[3] Markov_chain. (2016, November). Retrieved from https://en.wikipedia.org/wiki/Markov_chain
[4] Von Hilgers, Philipp, and Amy N. Langville. "The five greatest applications of Markov Chains." Proceedings of the Markov Anniversary Meeting, Boston Press, Boston, MA. 2006.
[5] Bikel, Daniel M., Richard Schwartz, and Ralph M. Weischedel. "An algorithm that learns what's in a name." Machine learning 34.1-3 (1999): 211-231.
[6] Leon-Garcia, Alberto, and Alberto. Leon-Garcia. Probability, statistics, and random processes for electrical engineering. Upper Saddle River, NJ: Pearson/Prentice Hall, 2008.
[7] California Indian Education. (2016, November). Retrieved from http://www.californiaindianeducation.org/inspire/world/
[8] L. Baum et. al. A maximization technique occuring in the statistical analysis of probabilistic functions of markov chains. Annals of Mathematical Statistics, 41:164–171, 1970.
[9] Cardie, Claire, and Raymond J. Mooney. "Guest editors' introduction: Machine learning and natural language." Machine Learning 34.1 (1999): 5-9.
[10] Marsland, Stephen. Machine learning: an algorithmic perspective. CRC press, 2015.
[11] Blunsom, Phil. "Hidden markov models." Lecture notes, August 15 (2004): 18-19.
[12] Collobert, Ronan, et al. "Natural language processing (almost) from scratch."Journal of Machine Learning Research 12.Aug (2011): 2493-2537.
[13] Bin, Tian, et al. "A Chinese word dividing algorithm based on statistical language models." Signal Processing, 1996., 3rd International Conference on. Vol. 1. IEEE, 1996.
[14] Leonardi, Riccardo, PierangeloMigliorati, and Maria Prandini. "Semantic indexing of soccer audio-visual sequences: a multimodal approach based on controlled Markov chains." IEEE Transactions on Circuits and Systems for Video Technology 14.5 (2004): 634-643.
[15] Sanderson, Conrad, and Simon Guenter. "On authorship attribution via Markov chains and sequence kernels." 18th International Conference on Pattern Recognition (ICPR'06). Vol. 3. IEEE, 2006.
[16] Lim, Bresley Pin Cheong, et al. "Web search with text categorization using probabilistic framework of SVM." 2006 IEEE International Conference on Systems, Man and Cybernetics. Vol. 4. IEEE, 2006.
[17] Velivelli, Atulya, and Thomas S. Huang. "Automatic video annotation using multimodal Dirichlet process mixture model." Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on. IEEE, 2008.

[18] Chen, Zhi-li, et al. "Effective linguistic steganography detection." Computer and Information Technology Workshops, 2008. CIT Workshops 2008. IEEE 8th International Conference on. IEEE, 2008.
[19] Dowman, Mike, et al. "A probabilistic model of meetings that combines words and discourse features." IEEE Transactions on Audio, Speech, and Language Processing 16.7 (2008): 1238-1248.
[20] Li, Lishuang, Zhuoye Ding, and Degen Huang. "Recognizing location names from Chinese texts based on max-margin markov network." Natural Language Processing and Knowledge Engineering, 2008. NLP-KE'08. International Conference on. IEEE, 2008.
[21] Meng, Peng, et al. "Linguistic steganography detection algorithm using statistical language model." Information Technology and Computer Science, 2009. ITCS 2009. International Conference on. Vol. 2. IEEE, 2009.
[22] Baomao, Pang, and Shi Haoshan. "Research on improved algorithm for Chinese word segmentation based on Markov chain." Information Assurance and Security, 2009. IAS'09. Fifth International Conference on. Vol. 1. IEEE, 2009.
[23] Wu, Ou, et al. "Rank aggregation based text feature selection." Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on. Vol. 1. IET, 2009.
[24] El Hassani, Ibtissam, AbdelazizKriouile, and Youssef BenGhabrit. "Measure of fuzzy presence of descriptors on Arabic Text Mining." 2012 Colloquium in Information Science and Technology. IEEE, 2012.
[25] Haji, Mehdi, et al. "Statistical Hypothesis Testing for Handwritten Word Segmentation Algorithms." Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on. IEEE, 2012.
[26] Zhou, Xiang-Dong, et al. "Handwritten Chinese/Japanese text recognition using semi-Markov conditional random fields." IEEE transactions on pattern analysis and machine intelligence 35.10 (2013): 2413-2426.
[27] Segarra, Santiago, Mark Eisen, and Alejandro Ribeiro. "Authorship attribution using function words adjacency networks." 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013.
[28] Rodrigues, Erica, et al. "Uncovering the location of Twitter users." Intelligent Systems (BRACIS), 2013 Brazilian Conference on. IEEE, 2013.
[29] Goyal, Anil, Mukesh K. Jadon, and Arun K. Pujari. "Spectral approach to find number of clusters of short-text documents." Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2013 Fourth National Conference on. IEEE, 2013.
[30] Shen, Jau Ji, and Ken Tzu Liu. "A Novel Approach by Applying Image Authentication Technique on a Digital Document." Computer, Consumer and Control (IS3C), 2014 International Symposium on. IEEE, 2014.
[31] Ahmed, Al-Falahi, et al. "Authorship attribution in Arabic poetry." 2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA). IEEE, 2015.
[32] Kirchhoff, Katrin, Sonia Parandekar, and Jeff Bilmes. "Mixed-memory Markov models for automatic language identification." Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on. Vol. 1. IEEE, 2002.
[33] Osiek, Bruno Adam, Geraldo Xexéo, and Luis Alfredo Vidal de Carvalho. "A language-independent acronym extraction from biomedical texts with hidden Markov models." IEEE Transactions on Biomedical Engineering 57.11 (2010): 2677-2688.
[34] He, Miantao, Miao Li, and Lei Chen. "Mongolian Morphological Segmentation with Hidden Markov Model." Asian Language Processing (IALP), 2012 International Conference on. IEEE, 2012.
[35] Alma'adeed, Somaya, Colin Higgens, and Dave Elliman. "Recognition of off-line handwritten Arabic words using hidden Markov model approach." Pattern Recognition, 2002. Proceedings. 16th International Conference on. Vol. 3. IEEE, 2002.
[36] Park, Hee-Seon, and Seong-Whan Lee. "Off-line recognition of large-set handwritten characters with multiple hidden Markov models." Pattern Recognition 29.2 (1996): 231-244.
[37] Espana-Boquera, Salvador, et al. "Improving offline handwritten text recognition with hybrid HMM/ANN models." IEEE transactions on pattern analysis and machine intelligence 33.4 (2011): 767-779.
[38] Imani, Zahra, et al. "offline Handwritten Farsi cursive text recognition using Hidden Markov Models." Machine Vision and Image Processing (MVIP), 2013 8th Iranian Conference on. IEEE, 2013.

[39] Prasad, Rohit, et al. "Improvements in hidden Markov model based Arabic OCR." Pattern Recognition, 2008. ICPR 2008. 19th International Conference on. IEEE, 2008.
[40] Zhou, GuoDong, and Jian Su. "Named entity recognition using an HMM-based chunk tagger." proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2002.
[41] Xu, Rong, et al. "Combining Text Classification and Hidden Markov Modeling Techniques for Structuring Randomized Clinical Trial Abstracts." AMIA. 2006.
[42] Yi, Kwan, and JamshidBeheshti. "A hidden Markov model-based text classification of medical documents." Journal of Information Science (2008).
[43] Frasconi, Paolo, Giovanni Soda, and Alessandro Vullo. "Hidden markov models for text categorization in multi-page documents." Journal of Intelligent Information Systems 18.2-3 (2002): 195-217.
[44] Vogel, Stephan, Hermann Ney, and Christoph Tillmann. "HMM-based word alignment in statistical translation." Proceedings of the 16th conference on Computational linguistics-Volume 2. Association for Computational Linguistics, 1996.
[45] Schuller, Björn, Gerhard Rigoll, and Manfred Lang. "Hidden Markov model-based speech emotion recognition." Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on. Vol. 2. IEEE, 2003.
[46] Nwe, Tin Lay, Say Wei Foo, and Liyanage C. De Silva. "Speech emotion recognition using hidden Markov models." Speech communication 41.4 (2003): 603-623.
[47] Juang, Biing Hwang, and Laurence R. Rabiner. "Hidden Markov models for speech recognition." Technometrics 33.3 (1991): 251-272.
[48] Gales, Mark, and Steve Young. "The application of hidden Markov models in speech recognition." Foundations and trends in signal processing 1.3 (2008): 195-304.
[49] Conroy, John M., and Dianne P. O'leary. "Text summarization via hidden markov models." Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2001.
[50] Maskey, Sameer, and Julia Hirschberg. "Summarizing speech without text using hidden markov models." Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers. Association for Computational Linguistics, 2006.
[51] Kupiec, Julian. "Robust part-of-speech tagging using a hidden Markov model." Computer Speech & Language 6.3 (1992): 225-242.
[52] Thede, Scott M., and Mary P. Harper. "A second-order hidden Markov model for part-of-speech tagging." Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. Association for Computational Linguistics, 1999.