Faculty of Mathematics
and Computer Science

Knowledge Based Systems and Language Technology Course

Applications of Hidden Markov Models in Natural Language Processing

Liviu-Ștefan Neacșu-Miclea

Department of Computer Science, Babeş-Bolyai University
1, M. Kogalniceanu Street, 400084, Cluj-Napoca, Romania
E-mail: liviu.neacsu@stud.ubbcluj.ro

Abstract

This report is a review of some applications of Hidden Markov Models (HMMs) in Natural Language Processing (NLP) contexts, with focus on three related areas: handwritten character recognition, named entity recognition, and machine translation. Three seminal research papers are analyzed in order to reveal the importance of these application areas, the specific HMM-based techniques employed, and the reported experimental results. Krevat and Cuzzillo (2006) describe an approach for off-line handwritten character recognition using HMMs, comparing its performance against a Naive Bayes classifier and a dictionary-based approach with near perfect results on a low scale dataset. Morwal et al.'s (2012) work studies the use of standard HMMs for language-agonstic named entity recognition. Last, but not least, Horiguchi et al. (2003) propose a new machine translation method using HMMs to learn translation template rules. A comparative analysis is performed on the methodologies and results presented in these papers, highlighting the flexibility and historical importance of HMMs in tackling various NLP challenges. While more advanced techniques have since emerged, this report emohasizes the fundamentalitly, as well as the continued relevance of HMMs in certain contexts and architectures.

Keywords:
Natural Language Processing, Handwitten Character Recognition, Named Entity Recognition, Machine Translation, Hidden Markov Model

1. Introduction

Hidden Markov Models (HMMs) are a significant class of statistical models specialized to handle sequential data, especially in scenarios where the process behind generating the observed data is not directly observable. Fundamentally, HMMs consist of a series of hidden states that drive the behavior of a system, and observable outputs, or emissions, probabilistically linked to these hidden states [9]. The model runs according to several key components: the sets of hidden states and possible observations, the transition probabilities between these hidden states, the emission probabilities of each observation from each hidden state, and the initial probabilities for starting in each hidden state [9]. The Markov property is a fundamental assumption in HMMs, which states that the next state of the system only depends on its current state, no matter what sequence of events comes before it [9]. Using this property, the model's complexity is reduced while still allowing it to capture temporal dependencies in data. HMMs are useful for modeling systems with latent processes that influence visible outcomes, making them suitable for a variety of real-world applications where the internal process cannot be directly investigated [9].

HMMs find their importance in diverse applications across numerous fields. In the domain of pattern recognition, HMMs have been used in tasks such as speech recognition, where the underlying phonemes or words are hidden, and the observed data is the acoustic signal [9]. Their acknowledged ability to handle temporal sequences has also made them naturally valuable in handwriting recognition, where the sequence of visual features like pen strokes can be modeled to infer the intended text [3]. In the field of natural language processing, HMMs have found applications in part-of-speech (POS) tagging, where the hidden states are the grammatical tags, and the observations are the words in a sentence [9] [2]. They are also used in named entity recognition (NER), helping in extracting and classifying entities such as names and locations within a text [2] [16] [13]. In machine translation, word sequences in one language are modeled to predict a suitable translated sequence in another [8].

Bioinformatics also benefit from the versatility of HMMs, in tasks like gene prediction, identifying coding and non-coding regions in DNA sequences [15], protein structure prediction, or finding the three-dimensional structure of proteins from their amino acid sequences [4]. In addition, HMMs are also employed in finances to predict market trends and in time series analysis to reveal patterns in temporal data [2]. Therefore, HMMs have become popular due to their great capacity to handle time-dependent patterns and probabilistic relationships, which makes them remarkably flexible for analyzing data that evolves over time.

This report starts with motivating the importance of the selected NLP topics. Then, a short section is dedicated to recent approaches in the targeted fields, followed by a section recalling a formal definition of HMMs and summaries of three seminal papers emplying HMM techniques in Handwritten Character Recognition (Kreval and Cuzzillo, 2006) [11], Named Entity Recognition (Morwal et al., 2012) [13] and Machine Translation (Horiguchi et al., 2003) [8]. A comparative discussion follows centered around these three approaching, highlighting particularities of each methodology and commenting the results and relevance. Finally, a conclusive section resumes the findings of this review and proposes possible extensions of this work.

2. Motivation

Offline handwritten character recognition, defined as converting handwritten text into its corresponding sequence of characters, better understood by the machine, finds significant importance in a number of applications, including digitizing historical texts for archival and accessibility [10]. Accuracy in recognition of handwritten characters is a reliable factor in automation in postal services and form processing [10]. This technology also helps individuals with disabilities in interacting with digital systems [10]. The difficulty of offline handwritten text recognition is due to the plentitude of variations in writing styles and potential noise in scanned images [1]. While early approaches utilized Hidden Markov Models (HMMs) [10] [12] [11] [3], current state-of-the-art techniques typically employ Deep Neural Network-based architectures [1], which are recognition methods praised for their robustness and performances [1].

Named Entity Recognition (NER) constitutes a fundamental task in Information Extraction and Natural Language Processing, and aims to identify and classify entities such persons, locations and organizations [16]. Attaining a good performance in NER is crucial for subsequent NLP applications including text summarization, machine translation, information retrieval or question answering [16]. HMMs have been investigated as a language-independent approach for NER [16] [13] [6] [17].

In an increasingly interconnected world, machine translation (MT) plays a great role in enabling inter-language communication. The field has seen an evolution from rule-based systems to statistical machine translation (SMT) and is currently dominated by neural machine translation (NMT), which has significantly improved translation accuracy [8]. For example, Google Translate's advancements confirm the impact of NMT. Despite this astounding progress, machine translation still cannot fully capture the nuances of different languages, and mitigating this issue remains a key challenge in machine translation.

3. Related work

Valuable insight into the current state of the art in handwritten character recognition, named entity recognition and machine translation can be gathered from recent literature surveys and reviews [1].

The field of handwritten character recognition is mainly lead by Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), however, HMMs are still relevant, especially in hybrid systems or for specific applications and are still a topic of on-going research in offline HTR [11] [7].

For named entity recognition tasks, HMMs were significant in the early days [16] [13]. Nevertheless, more recent approaches focused on using Conditional Random Fields (CRFs) and deep learning models such as LSTMs and Transformers, notably because they often outperform traditional HMMs, particularly on large datasets. However, HMMs are still a useful base model and can be efficiently applied in use cases with limited resource constraints or in combination with other techniques [6] [17].

In machine translation, the neural sequence-to-sequence models with attention mechanisms revolutionized the field. While HMMs were a core player in earlier statistical machine translation systems [8], they are less relevant in the current state-of-the-art. Despite their obsolescence, HMMs set in stone the fundamental ideas of probabilistic sequence modeling which is still a key factor in developing more complex models [9] [2].

Over the years, contributions involving HMMs were proposed in domains such as speech recognition [10], from which the paradigm extended to other sequence modeling tasks like OCR and handwriting recognition [12]. Moreover, hybrid HMM/ANN models were explored, trying to make use of both statistical understanding of HMMs and connection power of netural networks in order to enhance recognition accuracy [7].

These underlying works highlight the historical importance of HMMs and their impact in the evolution of techniques in these areas.

4. Reviewing the applications of HMM in Natural Language Processing

4.1. Hidden Markov Models

A Hidden Markov Model (HMM) is a statistical model that describes a system assumed to be a Markov process with unobserved (hidden) states [9]. An HMM is formally defined by a tuple $\lambda = (S, O, A, B, \pi)$, where S is a set of N hidden states $\{S_1, S_2, \ldots, S_N\}$, O is a set of M possible observations $O_1, O_2, \ldots, O_M$, A is an $N \times N$ transition probability matrix where $A_{i,j} = P(q_{t+1} = s_j \mid q_t = s_i)$ denotes the probability of transitioning from hidden state s_i to hidden state s_j at time $t + 1$ in a state sequence $(q_t)_{t=\overline{1,T}}$, B is an $N \times M$ emission probability matrix where $B_{i,k} = b_i(k) = P(o_k \mid q_t = s_i)$ represents the probability of observing output o_k given that the current hidden state at time t is s_i, and $\pi = [\pi_1 \ldots \pi_N]$ is an initial state probability vector, where $\pi_i = P(q_1 = s_i)$ represents the probability of starting in hidden state s_i at time $t = 1$ [9].

The fundamental principle behind HMMs is the Markov property, which asserts that the future state of a system only depends on its current state, and not on the events history that preceded it [9]. This "memoryless" property simplifies the model while still allowing for the capture of sequential dependencies. Three core problems are associated with HMMs: evaluation, decoding, and learning [9].

The evaluation problem consists of calculating the probability of an observed sequence given an HMM, $P(O \mid \lambda)$ [9]. It can be efficiently solved using dynamic programming approaches such as the forward and backward algorithms [9]. These algorithms compute the probability by summing over all possible hidden state sequences that could have generated the observed sequence [9].

The aim of the decoding problem is to find the most likely sequence of hidden states $Q = \{q_1, q_2, \ldots, q_T\}$ that produced a given observation sequence $O = \{o_1, o_2, \ldots, o_T\}$ [9]. An efficient solution to this problem is the Viterbi algorithm, which works by finding the single best path of hidden states through the model [9]. The learning task implies parameters estimation for A, B, and π of the HMM provided a set of observed sequences [9]. There exists an iterative procedure to find the likelihood estimates of these parameters, namely the Baum-Welch algorithm, which is a specific case of the Expectation-Maximization (EM) algorithm [9]. This algorithm starts with an initial guess for the model parameters and iteratively optimizes them in order to maximize the probability of the observed data [9].

HMMs have been particularly successful in sequence labeling tasks within NLP. These tasks involve assigning a label to each element in a sequence of observations. For example, in Part-of-Speech (POS) tagging, the observed sequence is a sentence, seen as a sequence of words, and the hidden states represent the POS tags (e.g., noun, verb, adjective) for each word. The HMM learns the probabilities of transitioning between different POS tags and the probabilities of each word being associated with a particular tag [9] [2]. Similarly, in Named Entity Recognition (NER), the observed sequence is a sentence, and the hidden states correspond to the named entity tags (e.g., person, location, organization) for every word. The HMM can then be used to predict the most likely sequence of named entity tags for a given sentence [16] [13] [6] [17]. The ability of HMMs to capture the local dependencies between elements in a sequence, as well as their probabilistic nature, makes them well-suited for these types of NLP tasks.

4.2. HMM in Handwritted Character Recognition

The work of Krevat and Cuzzillo (2006) aims to measure how a HMM which models correlations between adjacent characters improve the recognition accuracy [11]. The study tries to create an effective classifier that fits a specific dataset of handwritten words and benchmarks its performance against simpler methods [11]. The HMM approach is compared with a baseline Naive Bayes classifier, which treats each pixel independently, and with a a dictionary creation and lookup algorithm that holds the specific features of their dataset [11]. The scope of this comparison is to assess the benefits of employing HMMs for this task by putting them against both a fundamental probabilistic method and a specialized technique that well fits the dataset.

As HMM architecture is concerned, the hidden states are the letters corresponding to each segmented character image (pre-segmentation is assumed in the process), and the observed variables are the pixel bitmaps of these images [11] — as seen in Figure 1. A first-order HMM was mainly analyzed, where the probability of a character depends only on the preceding character [11]. The authors have considered using a second-order Markov assumption to potentially capture more complex relationships, but found it computationally expensive due to the substantial increase in the number of most probable paths that needed to be tracked during the Viterbi algorithm [11]. For the emission probabilities, the probability of observing a particular pixel configuration given a character state was computed using a Naive Bayes approach applied over each pixel of the character image [11]. This choice was made in favor of extracting more complex global features from images, due to the relatively low resolution of their dataset compared to other handwritten datasets [11]. This way, an equilibrium is achieved between the sequential dependencies capturing power and computational costs in training and inference.

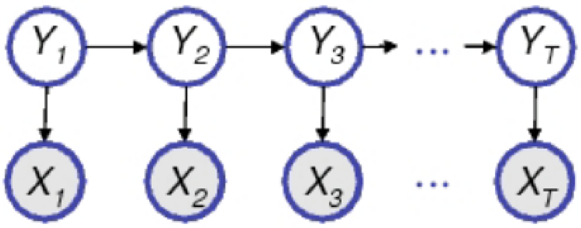

Fig. 1: HMM architecture for handwritten character recognition, where hidden variables Y_t are Latin letters and observed states X_t are bitmap images [11]

Experimentally, two versions of their HMM algorithm were tested [11]. A key optimization was implemented in the Viterbi algorithm, which is used to find the most likely sequence of hidden states (characters) given the observed images. This optimization consisted in eliminating the influence of character transitions for the last characters of each word [11]. The reasoning behind this idea was to let the model focus on learning the more strongly correlated dependencies between characters within a word, rather than being influenced by the often noisy and uncorrelated transitions between the end of one word and the beginning of the next one [11]. The comparison with the baseline Naive Bayes classifier proved that the HMM achieved a higher accuracy (71%), justifying the benefit of considering the contextual information of adjacent characters [11]. Remarkably, the dictionary creation and lookup algorithm, whose feasibility is due to the small size of the dataset (55 distinct words), yielded almost perfect results in classification [11]. However, the authors acknowledged that this approach is not scalable to larger datasets [11]. The findings of this study reveal that while HMMs are able to improve handwritten character recognition by taking advantage of correlations between characters, their performance can be influenced by the specific characteristics of the dataset, and specialized methods might prove more suitable for certain use cases.

4.3. HMM in Named Entity Recognition

The work of Morwal, Jahan, and Chopra (2012) focused on applying HMMs to the task of named entity recognition (NER), particularly accentuating its potential for languages like Hindi and Urdu where capitalization, which usually makes NER easier in languages like English, is absent [16]. This work introduces in detail a language-agnostic approach based on HMM for NER, with the mention that it can be adapted to other variety of linguistic contexts [13].

The proposed method does not hard-code the states of HMM, instead they are dynamically found according to the named entity tags of interest, allowing for flexibility in choosing the types of entities that can be recognized [13]. Their methodology employs a learning by example paradigm, meaning that the HMM parameters are estimated from an annotated corpus, therefore adapting the system to a different language requires little to no effort on the human side [13]. The aim of this approach is to establish a generic technique applicable to different languages, addressing the issues raised by the lack of capitalization or scarcity of training resources in some languages. The standard definition of an HMM was used, with its transition (A), emission (B) and start (π) probabilities [13]. In this context, the hidden states are represented by the named entity tags be it a person name, location, or organization name. The set of named entity types is obtained from an annotated training dataset [13]. The observed variables are the words sequence of the input text [13]. The start probability π denotes how likely a sentence begins with a particular named entity tag [13]. The transition probability A captures the probability of switching from one named entity tag to another in the sequence [13]. The emission probability B represents the probability of observing a specific word given a certain named entity tag [13]. The Viterbi algorithm was used to identify the most likely sequence of named entity tags for an input sentence [13]. By seeing the NER problem as an HMM sequence labeling task, the model can learn the statistical patterns of named entities appearances in sequence and the likelihood of words belonging to certain entity categories.

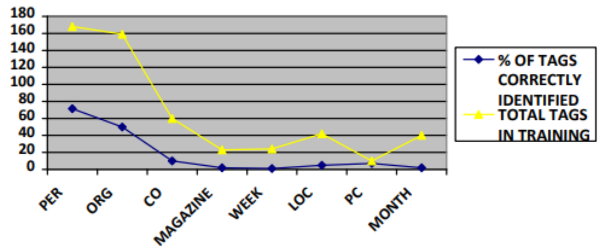

Fig. 2: Accuracy of HMM in NER for English language [5]

While on specific experimental results are provided, the what maked HMMs in NER distinguish from other approaches is model's ability to capture the local context of named entities within the text [13]. The ease of understanding, implementation and scalability to large datasets are properties which make it a practical choice for various applications [13]. Moreover, another study of HMM in NER featuring Dari - a language that also lacks capitalization - achieved an accuracy of 94%, confirming the potential of this approach in handling the specific challenges of these languages [17]. An English particularization of the language-agnosted approach was conducted by Chopra and Morwal with over 70% accuracy and an F-Measure of 73.8% [5] — detailed results in Figure 2. The performance of NER systems is typically evaluated using classification metrics such as Precision, Recall, and F-measure, which quantify the ability of the system to identify and classify the named entities [6]. Therefore, the discussion and similar works point to the fact that HMMs can be a valuable tool for named entity recognition, especially for their language independence feature and ability to model sequential dependencies.

4.4. HMM in Machine Translation

Horiguchi, Shimazu and Nguyen (2003) address in their work the continuous challenges in machine translation, especially the limitations of traditional statistical methods in capturing contextual nuances and managing structural variations across languages [8]. An innovative translation method is proposed to overcome these issues, blending the principles of Example-Based Machine Translation (EBMT) with the probabilistic framework of Hidden Markov Models (HMMs) [8]. The main idea is to first learn translation templates from a bilingual corpus and then use an HMM to find the most probable sequence of these templates when translating a given source language sentence [8]. This integration enhances translation accuracy while also reducing computational complexity that earlier template learning techniques suffer from [8].

Example-Based Machine Translation (EBMT), which stays at the foundation of this research, is a paradigm that interprets translation as a problem of recognizing analogies [8]. The model translates a sentence in the source language by comparing it against a database of previously pairs of sentence and its translation (which is called a bilingual parallel corpus) in order to find the most similar samples [8]. Then, the translations of these related source sentences are then adapted and recombined to create a translation for the new input sentence [8]. To start with, this process identifies similarities and differences between the source and target sentences within the examples dataset. Two sentences sharing a continuous sequence of identical root words or morphemes are said to have a similarity, while a difference appears when a subsequence from one sentence has nothing in common with a subsequence from the other sequence, but they are equivalent in terms of translation patterns [8]. For example, the sentences "I bought the book for John" and "I bought the ring for John" have the similarities "I bought the" and "for John", while "book" and "ring" are aligned differences. The same reasoning holds for the translated Vietnamese counterparts - the language used in this research [8]. These results are arranged into a mathing sequence such as S_0^1 = "I bought the", D_0^1 =("book", "ring"), S_1^1 ="for John", and their corresponding translations S_0^2, D_0^2, S_1^2 [8].

Once these matching sequences are identified, two learning heuristic designed to pinpoint corresponding similarities (Similarity Translation Template - STTL) or differences (Difference Translation Template - DTTL) are used to infer translation templates [8]. The STTL works by replacing the found differences with variables, while DTTL replaces similarities with variables. Combining these two heuristics results in the template learning algorithm (TTL) [8]. Once learned, these translation templates are sorted based on their specificity - the sentence with a higher number of terminals (tokens that are not variables) are said to be more specific [8].

Based on the previously matched sequences with variables, template rules are defined as $S_1S_2 \ldots S_n \leftrightarrow T_1T_2 \ldots T_k$, where S_i denotes a words sequence or a variable from the source language, and T_i is the corresponding translation in the target language, also a constant sequence of words or a variable [8]. It is important to know which source variables correspond to which target variables, especially due to property of some languages to allow entangled structures which may place some variables out of order, like for example, the Vietnamese translation is the English phrase "give [X] up" has an atomic target sequence for the similarity part, "từ bỏ [X]" [8].

The authors observe that the template learning approach grows exponentially in complexity with the length of the input sentence and the number of template rules, especially ones with multiple variables [8]. Identifying all applicable rules and combining them to generate a translation can therefore become a demanding task [8]. This greatly justifies the opportunity to consider including HMMs into the core of an innovative methodology [8]. The template learning translation problem finds a reformulation into an equivalent problem compatible with the HMM foundations [8]. The input sentence is sliced into various possible sequences of substrings, with each substring potentially matching the left-hand side of one or more template rules [8].

The HMM model (Figure 3) identifies the most probable sequence of lexical rules that can be applied to the input sentence to produce the translated output [8]. A lexical rule is defined as a template rule devoid of any variables, and can be considered a hidden state within the HMM [8]. Additionally, the observed symbol emmited by the state is represented by an input sequence substring that matches the source language side of that lexical rule [8]. Therefore, the translation problem becomes a task of finding the most probable sequence of hidden states (lexical rules) that could have generated the observed sequence of input substrings [8]. This matches the formal specification of a HMM model. The objective is to maximize the probability $P(r_i|e_2, e_2, \ldots, e_m)$, where r_i denotes a sequence of lexical rules and $e_1, \ldots, e_m$ are the observed substrings of the input sentence [8]. The work assumes a Markovian process in order

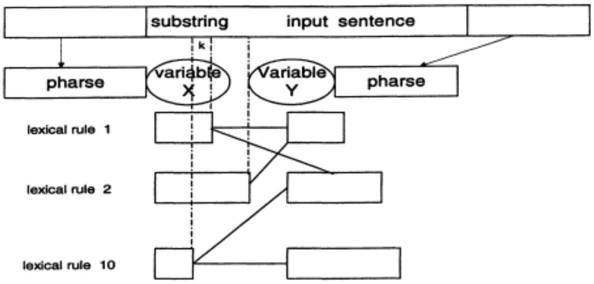

Fig. 3: Translation HMM architecture [8]

to simplify the process, and the Viterbi algorithm is used for finding the most likely sequence of lexical rules, thus bringind down the complexity to polynomial time by this dynamic programming approach [8].

Naturally, from this HMM description it follows that transition probabilities are the likelihood of moving from one hidden state (lexical rule) to another in the sequence of applied rules, capturing the sequential dependencies between the usage of difference basic translation units [8]. Emission probabilities show how likely it is to observe a certain substring in the input sentence assuming that the HMM is in a specific hidden state, meaning that a certain lexical rule is matched [8]. The initial state probabilities specify the probabilities of the process starting in a particular hidden state, the lexical rule which is applied at the beginning of the input sentence [8].

The Forward-Backward algorithm is used for the parameter estimation step of the HMM model [8]. The bilingual corpus feeds pairs of source and target sentences as training data together with a common set of template rules [8].

The model learns the transition probabilities between lexical rules and emission probabilities from lexical rules to the observations [8]. These probabilities are initialized based on frequency counts from the corpus [8]. In other words, the transition probability between lexical rules is approximated as the relative frequency of one rule following another, and the emission probability represents the frequency of an observed symbol given a certain rule [8].

By effectively integrating Hidden Markov Models with the learning of translation templates, the authors address the computational limitations of traditional template-based approaches and enhance the accuracy of translations. The proposed method, which frames translation as finding the most probable sequence of lexical rule applications, offers a robust and adaptable framework for handling the complexities of natural language. The experimental results, particularly in the context of Vietnamese sentence reduction, provide evidence of the effectiveness of this hybrid approach, demonstrating its ability to potentially outperform other baseline methods and handle challenging cases . This research highlights the potential of leveraging statistical sequence modeling techniques to further refine and improve the quality and efficiency of example-based machine translation systems. Future research could explore different HMM architectures , incorporate more sophisticated linguistic features , and extend the application of this method to a wider variety of language pairs , potentially leading to even more significant breakthroughs in the field.

The experiments were run on a corpus of 1200 English-Vietnamese translated pairs, resulting in a total of 11034 template rules and 2287 lexical rules [8]. The reported experimental outcomes of this research proved that the proposed HMM-based translation approach significantly enhances translation accuracy (from 34% the state of the art at that time to 81% with HMM) while also drastically improving the computational costs with the help of the TTL algorithm [8]. This makes the mothod it more suitable for real-world applications [8]. Additionally, EBMT-HMM has also proved useful in other NLP tasks, such as sentence reduction, which implies shortening an input sentence in such a way that it drops the redundancies [14].

5. Discussion

The three works analyzed in this report prove that Hidden Markov Models are a flexible tool in addressing various problems which require identifying patterns in natural language processing. While each application is based on the fundamental principles of HMMs, the specific techniques and the evaluation metrics employed are dependent on the characteristic challenges of each domain.

	Krevat & Cuzzillo (2006) [11]	Morwal, Jahan, & Chopra (2012) [13]	Horiguchi, Shimazu, & Nguyen (2003) [8]
Task	Off-line handwritten character recognition	Named entity recognition	Translation template learning for machine translation
HMM Techniques	First-order HMM, Naive Bayes for emission probabilities over pixels, Viterbi optimization ignoring inter-word transitions	Language-agnostic HMM with dynamic states based on NE tags, learning by example from annotated corpus, Viterbi decoding	HMM for learning lexical translation rules (hidden states), input substrings as observations, Viterbi algorithm for template application
Evaluation Metrics	Accuracy (compared to Naive Bayes and dictionary lookup)	Precision, Recall, F-Measure [5]	Improvement in translation accuracy, low computational complexity, outperformed TTL in sentence reduction

Table 1: Summary of Methodologies

As showcased in Table 1, all three studies use HMMs for sequence modeling applications. However, the definition of states and observations depends on the actual problem which is solved. In handwritten character recognition, the hidden states are individual characters, and the observations are the pixel data of the character images. In named entity recognition, the hidden states represent the named entity tags, and the observations are the words in the text. In the case of translation template learning, the hidden states are the lexical translation rules, and the observations are the substrings from the input sentence.

The Viterbi algorithm is a common point in all applications. In handwritten recognition and NER, both involve finding the most likely sequence of hidden states given a sequence of observations, which is characteristic of sequence labeling tasks [11] [13]. In machine translation, Viterbi algorithm is used to find the output translations only after learning the translation patterns and templates [8].

The evaluation of the models slightly differs accross the three approahes, but they share the usual classification metrics: accuracy, precision, recall and F-Measure. The machine translation approach also manages to bring the computational time complexity into the polynomial range, outperforming other solutions from that time.

Together, these studies reveal the strength of HMMs in dealing with contextual dependencies in sequential data, proof being improved accuracy in handwritten recognition and machine translation, and adaptability to different languages for the NER task. The HMMs are able to learn patterns and reuse them. One possible limitation could be the Markov assumption, which, despite its usefulness in bringing down the complexity of the model, comes at the cost of limiting the model's learning capacity. This is the reason some researchers may focus their attention to other improvements such as the second-order HMMs, or on the contrary, impose even more limits in situations where information redundancy is inherently proved [11].

It is also worth mentioning that HMMs were a leading technique in these fields, until the significant advances of deep learning occurred and became a mainstream topic. Nowadays, HMMs are still used combined with neural networks or have been replaced by them in many state-of-the-art solutions [3].

6. Conclusions and future work

The proof stands still that Hidden Markov Models have been a remarkable and influential innovation featuring flexibility and making it possible to addressing complex sequence modeling problems across a wide range of disciplines, from natural language processing and not only.

The analysis of the three works in this report reveals the way HMMs can be applied to tackle specific challenges in off-line handwritten character recognition, named entity recognition, and machine translation. In every case, the HMM model demonstrated their ability to model underlying hidden processes that generate observable sequences, providing precious insight into the data in an efficient manner. The benefits of HMM surpasses its implementation efforts compared to simpler, but weaker approaches, opening the way to adaptability event with limited data and nurturing an impressive boost in accuracy. While deep learning has become dominant in these fields, the historical importance of the fundamental principles of HMMs in shaping the current course of progress is not to be neglected. They even still see their use in hybrid systems or specialized applications which enhance their enduring relevance. When limitations such as small data or computational resources become a problem, or when interpretability of the model is crucial, one can still considered HMMs as a powerful and effective approach to sequence modeling. Further research may continue to explore new ways to integrate HMMs with more recent techniques to make use of their strengths in sequence modelling fields.

References

[1] AlKendi, W., Gechter, F., Heyberger, L., Guyeux, C., 2024. Advancements and challenges in handwritten text recognition: A comprehensive survey. Journal of Imaging 10, 18.

[2] Almutiri, T., Nadeem, F., et al., 2022. Markov models applications in natural language processing: a survey. Int. J. Inf. Technol. Comput. Sci 2, 1–16.

[3] B, T.R., 2024. Handwritten character recognition system. Journal of Electrical Systems 20, 1465–1475. URL: http://dx.doi.org/10.52783/jes.3553, doi:10.52783/jes.3553.

[4] Choo, K.H., Tong, J.C., Zhang, L., 2004. Recent applications of hidden markov models in computational biology. Genomics, proteomics & bioinformatics 2, 84–96.

[5] Chopra, D., Morwal, S., 2013. Named entity recognition in english using hidden markov model. International Journal .

[6] Chopra, D., Morwal, S., Purohit, G., 2013. Hidden markov model based named entity recognition tool. International Journal in Foundations of Computer Science & Technology (IJFCST) 3, 67–73.

[7] Espana-Boquera, S., Castro-Bleda, M.J., Gorbe-Moya, J., Zamora-Martinez, F., 2010. Improving offline handwritten text recognition with hybrid hmm/ann models. IEEE transactions on pattern analysis and machine intelligence 33, 767–779.

[8] Horiguchi, S., Shimazu, A., Nguyen, M.L., 2003. Translation template learning based on hidden markov modeling, in: Language, Information and Computation: Proceedings of the 17th Pacific Asia Conference, 1-3 October, 2003, Sentosa, Singapore, Waseda University. pp. 269–276.

[9] Jurafsky, D., Martin, J.H., 2018. Hidden markov models. Speech and language processing 3.

[10] Kannan, R.J., Prabhakar, R., Suresh, R., 2008. Off-line cursive handwritten tamil character recognition, in: 2008 International Conference on Security Technology, IEEE. pp. 159–164.

[11] Krevat, E., Cuzzillo, E., 2006. Improving off-line handwritten character recognition with hidden markov models. Transaction on Pattern Analysis and Machine Learning 33.

[12] Kundu, A., He, Y., Bahl, P., 1989. Recognition of handwritten word: first and second order hidden markov model based approach. Pattern recognition 22, 283–297.

[13] Morwal, S., Jahan, N., Chopra, D., 2012. Named entity recognition using hidden markov model (hmm). International Journal on Natural Language Computing (IJNLC) Vol 1.

[14] Nguyen, M.L., Horiguchi, S., Shimazu, A., Ho, B.T., 2004. Example-based sentence reduction using the hidden markov model. ACM Transactions on Asian Language Information Processing (TALIP) 3, 146–158.

[15] Yoon, B.J., 2009. Hidden markov models and their applications in biological sequence analysis. Current genomics 10, 402–415.

[16] Zhou, G., Su, J., 2002. Named entity recognition using an hmm-based chunk tagger, in: Proceedings of the 40th annual meeting of the association for computational linguistics, pp. 473–480.

[17] Zia, G.A.J., Sharifi, A.Z., et al., 2019. Hmm-based dari named entity recognition for information extraction, in: CS & IT Conference Proceedings, CS & IT Conference Proceedings.

Acknowledgement: This work is the result of my own activity, and I confirm I have neither given, nor received unauthorized assistance for this work. I declare that I did not use generative AI or automated tools in the creation of content or drafting of this document.