I.J. Information Technology and Computer Science, **2022, 2, 1-16**
Published Online April 2022 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2022.02.01

Markov Models Applications in Natural Language Processing: A Survey

Talal Almutiri
Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia.
E-mail: almutiri.talal@hotmail.com

Farrukh Nadeem
Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia.
E-mail: fabdullatif@kau.edu.sa

Received: 09 August 2021; Revised: 09 October 2021; Accepted: 16 November 2021; Published: 08 April 2022

Abstract: Markov models are one of the widely used techniques in machine learning to process natural language. Markov Chains and Hidden Markov Models are stochastic techniques employed for modeling systems that are dynamic and where the future state relies on the current state. The Markov chain, which generates a sequence of words to create a complete sentence, is frequently used in generating natural language. The hidden Markov model is employed in named-entity recognition and the tagging of parts of speech, which tries to predict hidden tags based on observed words. This paper reviews Markov models' use in three applications of natural language processing (NLP): natural language generation, named-entity recognition, and parts of speech tagging. Nowadays, researchers try to reduce dependence on lexicon or annotation tasks in NLP. In this paper, we have focused on Markov Models as a stochastic approach to process NLP. A literature review was conducted to summarize research attempts with focusing on methods/techniques that used Markov Models to process NLP, their advantages, and disadvantages. Most NLP research studies apply supervised models with the improvement of using Markov models to decrease the dependency on annotation tasks. Some others employed unsupervised solutions for reducing dependence on a lexicon or labeled datasets.

Index Terms: Hidden Markov Models, Markov Chains, Named Entity Recognition, Natural Language Generation, Natural Language Processing, Parts of Speech Tagging, Quantitative Analysis.

1. Introduction

Linguistic science sets out to achieve the characterization and explanation of the multiplicity of linguistic signifiers in our environment, whether in the spoken word, in written form, or in some other medium. An element of this is related to the cognitive explanation of how humans manage the acquisition, production, and understanding of language; another element is related to how we understand how language relates to the world as a whole, while a third element relates to how communication is achieved in language through linguistic structure. To examine this third element, it has been suggested that a set of rules governs linguistic expression. This fundamental viewpoint has existed for at least the last two millennia. However, in the 20th and 21st centuries, linguistic examination has increased in both informality and rigor as linguists have created intricate grammatical models that explain both "good" and "bad" language usage. Statistical language models have been constructed and effectively employed in various domains of NLP. Although being useful in practice is not the same thing as developing a valid theory, the effectiveness of statistical language models appears to demonstrate that the fundamental approach is correct [1].

NLP is one of many disciplines that are reliant on the use of statistical modeling. Statistical NLP draws inferences from statistics to be used in NLP. This involves acquiring data that have been generated through an unknown probability distribution and drawing inferences from them. In terms of machine learning, among the considerable array of means to undertake natural language processing, the Markov model is a significant one[2].

A natural language is a probabilistic language that relies on a sequence of words to get a meaning within context; therefore, stochastic models such as Markov models are suitable for this purpose. The objective of this study was to summarize some of the current researches that introduced different solutions for employing Markov models in NLP with focusing on methods/techniques provided, contributions/advantages, and limitations/disadvantages. Therefore, interested researchers will find documented summarization about the recent studies that have been employed Markov Models in the NLP domain, and they will start where others end effortlessly.

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

To understand the Markov model, we require some fundamental information regarding random processes, stochastic processes, and deterministic processes. We also need to define the meaning of state, state-space, and general processes. A "state" comprises a collection of variables that are each assigned values, generally used to describe physical environments, e.g., we have the weather states of cloudy, rainy, or sunny. Process describes the movement between states; processes change states under the state-space that contains every potential state.

Deterministic processes follow a collection of formulas/equations representing exactly how systems will develop as time progresses. In stochastic processes, evolutionary processes have an element of randomness, and, if we repeat a process repeatedly, we will get several different outcomes. Opposing the stochastic model is the deterministic model [3], which contains a set of equations describing precisely the ways a system will evolve against time. When we run stochastic processes often, we will not achieve matching results; these various runs are frequently referred to as "realizations" of the stochastic model. It is generally easier to undertake analysis of a deterministic model than it is of a stochastic model.

Nevertheless, in numerous instances there is greater realism to a stochastic model, especially in problems involving "small numbers." As an example, we can imagine we are modeling how we can manage an endangered species, examining how a variety of strategies might influence its survival. A deterministic model would not be optimal in this case, as the equations will either predict that the species must become extinct or must survive. Stochastic models can encompass the probabilities of extension occurring.

A Markov model is a technique dealing with the likelihood of something happening in the future by analyzing the probabilities that we currently know. These models are widely used in the business world, particularly for analyzing market share, in meteorology, in education for predicting future student enrollment, and in manufacturing, for calculating the likelihood of machines failing at some future point [4].

As we mentioned, natural languages are probabilistic languages that depend on words and sentences order and sequence to gain meaning in context, stochastic models like Markov models are appropriate. According to the facts provided in this study, we also intend to determine the limits of previous studies in order to assist interested researchers in identifying research gaps or even employing the benefits of utilizing Markov models in their research projects. Without a doubt, this benefit will be followed by academic accomplishments, such as motivating people to write more scientific publications employing Markov models in the NLP field.

2. Markov Models

In Markov analysis, we assume an initial starting condition or state for the system, e.g., an initial state could be a pair of rival manufacturers, one with 40% of market share and the other 60%. Over time, market shares could alter to 45% and 55%. To predict such an outcome, we have to know how likely are the probabilities that the system will change from the former state to the latter. We can take all the probabilities in a specific problem and set them in a table or matrix. The matrix of transition probabilities indicates how likely it is that a system change will occur over time. This is how the Markov process works, enabling us to make predictions regarding a state or condition over time [5]. The section below will offer a descriptive outline of the two most commonly used Markov models, the Markov chain, and the Hidden Markov Model.

2.1. Markov Chains

Markov chains are essential elements of stochastic processes. They are employed for many purposes in a multiplicity of disciplines. Markov chains are stochastic processes that satisfy Markov properties, meaning that both past and future have independence if we know the present. If we know a process's current state, we do not need any more information about its previous state for optimal predictions regarding the future.

Markov chains are stochastic (continually changing) models employed to predict/estimate/guess the result of a specific event when we only know the previous state and what is happening in the present. By state we mean the conditions pertaining at a particular time. In a more formal model, if we have sequential variables for state $s_1, s_2, \ldots, s_i$. The Markov model will embody the assumption of Markov model regarding the sequences list of probabilities, i.e., for predictions of the future we only need to know the present, not the past.

The assumption of the Markov Model:

$$P(s_i = a \mid s_1 \ldots s_{i-1}) = P(s_i = a \mid s_{i-1}) \quad (1)$$

The formal specification of a Markov chain shown in Table 1:

Markov chains are employed to calculate the likelihood of events occurring by regarding them as a state that transitions into another state or back to its previous state. If we use the example of weather prediction shown in Figure 1, if we make a random selection of probabilities, we can say that, if it is sunny today, the chance to be rainy tomorrow will be 30%; but if the weather today is a rainy, there is a 20% chance the weather will be sunny in the next day. Thus, if it is sunny today, it is 70% likely the next day will be sunny; if it is raining today, there is an 80% probability that tomor

row will also be raining. We can summarize this using a transition diagram which describes every possible state transition [7]:

Table 1. Mathematical representation of Markov chains [6].

$\boldsymbol{S = s_1, s_2, \dots, s_N}$	A group of N states
$\boldsymbol{A = a_{11}\, a_{12} \dots\, a_{N1} \dots q_{NN}}$	A **transition probability matrix** A, every a_{ij} being a representation i the likelihood of changing from one state i to another state j $\sum_{i=1}^{n} a_{ij} = 1 \ \forall i$
$\boldsymbol{P = p_1, p_2, \dots, p_N}$	**Initial probability distribution** regarding states S. p_i Means the likelihood which the Markov chain will begin at a particular state i. Certain states j may equal zero $p_i = 0$, which means they may not represent initial states. Also, $\sum_{i=1}^{n} p_i = 1$

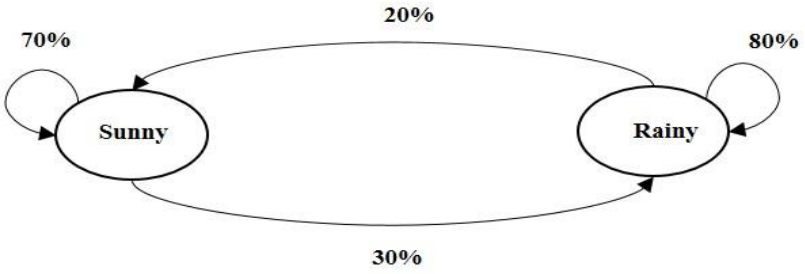

Fig.1. Weather prediction example using Markov chains [7].

2.2. Hidden Markov Model

The Hidden Markov Model (HMM) is a mathematical technique representing a substantial and helpful collection of stochastic processes. It can be identified through the mark of property, meaning that a process's future state is solely conditional on the current state and not the history. HMM was first formulated by Baum and Petrie [8] with its first and most important application being in automatic speech recognition [9]. Markov models have extremely high-order mathematical structures and, with proper application, they have important practical uses in several areas [10].

Markov chains are helpful when there is a requirement for computing probabilities in sequences of observable events. In numerous instances, the events of interest are not visible and cannot be directly observed: e.g., a human reader does not usually distinguish tags of part-of-speech when scanning a text, they draw an inference about the tags from the sequence of words. We refer to the tags as "hidden" as we do not directly observe them.

The HMM lets us analyze observed events, e.g., words that have been observed in a sentence or a whole text, as well as invisible events, e.g., part-of-speech tags. The elements of HMMs are described in Table 2 [6]:

Table 2. Mathematical representation of Hidden Markov Model [6].

$\boldsymbol{S = s_1, s_2, \dots, s_N}$	A group of N states
$\boldsymbol{A = a_{11}, \dots\, a_{ij}, \dots q_{NN}}$	A **transition probability matrix** A, every a_{ij} being a representation, i, of the likelihood of changing from one state i to another state j $\sum_{i=1}^{n} a_{ij} = 1 \ \forall i$
$\boldsymbol{O = o_1, o_2, \dots, o_t}$	A sequence of T **observations,** O, all of them taken from a specific vocabulary $V = v_1, v_2, \dots v_v$
$\boldsymbol{B = b_i(o_t)}$	A sequence of **observational likelihoods** (a.k.a. **emission probabilities**), all of them stating the probability of observations o_t being created from states i
$\boldsymbol{P = p_1, p_2, \dots, p_N}$	**Probability distribution** of states' S. p_i means the likelihood which the Markov chain will begin at a particular state i. Certain states j may equal zero $p_i = 0$, which means they may not represent initial states. Also, $\sum_{i=1}^{n} p_i = 1$

As an example is shown in Figure 2, for the weather guessing game (The Viterbi algorithm) [11] we can see how we can predict the state "weather," which is hidden, based on knowledge regarding a person's daily activity; we can try to predict the weather state (hidden) B by correlating it with activity (observed) A. The **states** are rainy and sunny; the **Start probabilities** are 0.4 for sunny and 0.6 for rainy, showing the weather state (we know that it is more likely to be rainy). The **transition probability** is the chances of the weather changing in the foundational Markov chain. In our example, if it is raining today, there is only a 30% likelihood that it will be sunny tomorrow. **Emission probability** is a representation of the likelihood that a person will undertake a specific activity on a given day: if the sun is shining, there is a 60% likelihood that he will go for a walk, if it is raining, there is a 50% likelihood that he will stay in and clean his apartment.

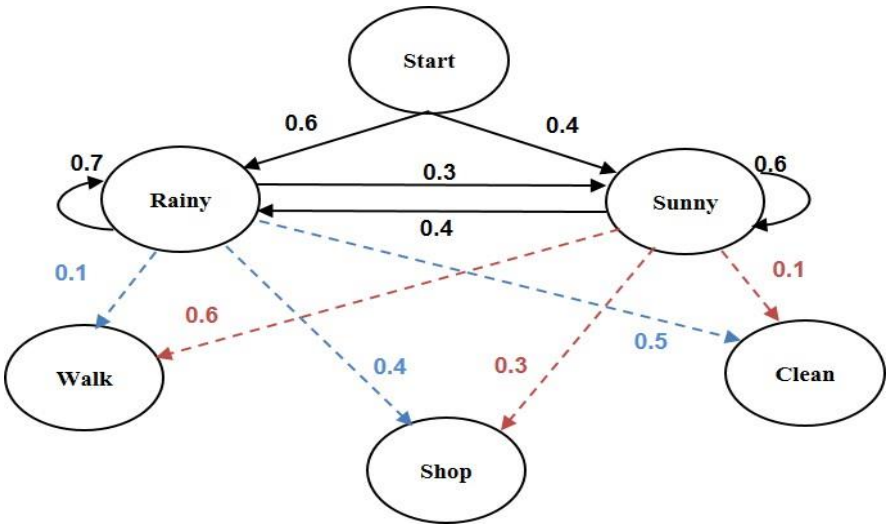

Fig.2. Weather guessing game example using Hidden Markov Model [12].

According to the example in Figure 2, Transition Probability will be as shown in Table 3, and Emission Probability will be represented as shown in Table 4:

Table 3. Transition probability for weather guessing game [12].

	The Next State	
The Current State	**Rainy**	**Sunny**
Rainy	0.7	0.3
Sunny	0.4	0.6

Table 4. Emission probability for weather guessing game [12].

	The Next State		
The Current State	**Walk**	**Shop**	**Clean**
Rainy	0.1	0.4	0.5
Sunny	0.6	0.3	0.1

As mentioned above, the transition probability is the probabilities of the weather changing in the foundational Markov chain. According to Table 3, there is a 40% chance of raining tomorrow if weather is sunny today. According to the Emission probability shown in Table 4, if the weather is sunny, there is a 10% chance that the person will stay in and clean his apartment, 30% he will go shopping, and 60% go for a walk. If it is raining, there is a 50% likelihood that he will stay in and clean his apartment, a 10% he will go for a walk, and a 40% likelihood of shopping.

3. Natural Language Processing

Natural Language Processing (NLP) is a section of Artificial Intelligence that relates to how the human language can be processed and understood. NLP [13] aims to find useful ways of reading, deciphering, understanding, and making sense of human language. From the 1950s onwards, how machines understand language has been central to entity extraction, information retrieval, document indexing, topic modelling, and translation. It is employed in modern computing to control search engines, detect spam messages, and increase the effectiveness of analytics in agile and scalable fashions. System performance has increased exponentially as computers have become more efficient, and machine learning has become more sophisticated. There are currently many NLP systems that may be regarded as operating at

close to human levels.

Natural Language Processing has five phases: Morphological and Lexical Analysis, Discourse Integration, Syntactic Analysis, Semantic Analysis, and Pragmatic Analysis [14]. In the following points, a brief introduction of each phase is given.

3.1. Morphological and Lexical Analysis

Language's lexicon contains a list of vocabulary with meaning and expressions that describe a language. Morphology analysis means discovering, describing the structure of words.

3.2. Syntactic Analysis.

This phase works to analyze the words in a sentence to depict the sentence's grammatical structure. The words are turned into a structure that demonstrates how they are related to one another.

3.3. Semantic Analysis.

It is about extracting the dictionary meaning based in the context, which tries to attach the meaning to the word's structure that was formed by the syntactic analyzer.

3.4. Discourse Integration.

Any single sentence's meaning is influenced by the phrases that precede it, as well as the meaning of the sentences that follow it.

3.5. Pragmatic Analysis.

It refers to the process of abstracting or deriving the deliberate use of language in situations, particularly ones in which world knowledge is required. The main focus is on what was said and how it is reinterpreted.

There are different applications of NLP, such as part-of-speech (POS) tagging, Text Summarization, Named entity recognition (NER), Machine translation, Questions Answered, and others. Also, there are various approaches to accomplish NLP tasks and applications such as Rules-based, Lexicon-Based, Stochastic approach, and Machine Learning-Based. A brief discussion of these approaches is provided in Section 5

4. Research Methodology

The current research was based upon the literature of application of Markov models in NLP from 2016 to 2020. We have collected 43 papers shown in Table 5 that focused on three domains of NLP: Natural Language Generation, Named-Entity Recognition, and Parts of Speech Tagging.

Table 5. Number of papers that collected from 2016 to 2020

Domain/ Year	2016	2017	2018	2019	2020	Total
Natural Language Generation	2	2	3	5	4	16
Named-Entity Recognition	2	3	3	4	2	14
Parts of Speech Tagging	3	2	2	2	4	13
Total	7	7	8	11	10	43

From the collected 43 papers, we have selected 18 papers that only related to Markov Chains and Hidden Markov Model. Table 6 shows the selected papers.

Table 6. The selected papers that applied Markov chains and HMM in NLP.

No	Authors	Year	Name of journal	Title
Natural Language Generation				
1	Zhang et al. [15]	2020	arXiv preprint	"Generating fluent adversarial examples for natural languages"
2	Martínez Garcia et al. [16]	2020	Sociedad Española para el Procesamiento del Lenguaje Natural	"A light method for data generation: a combination of Markov Chains and Word Embeddings"
3	Gehrmann et al. [17]	2019	arXiv preprint	"Improving human text comprehension through semi-Markov CRF-based neural section title generation"
4	Harrison et al. [18]	2017	Thirteenth Artificial Intelligence and Interactive Digital Entertainment Conference	"Toward Automated Story Generation with Markov Chain Monte Carlo Methods and Deep Neural Networks"
5	Yang et al. [19]	2018	arXiv preprint	"Automatically generate steganographic text based on Markov model and Huffman coding"
6	Luo et al. [20]	2016	Transactions on Internet and Information Systems (TIIS)	"Text Steganography Based on Ci-poetry Generation Using Markov Chain Model"

Named-Entity Recognition				
7	Miller et al. [21]	2020	International Workshop on Mining and Learning with Graphs (MLG)	"Collective Bio-Entity Recognition in Scientific Documents using Hinge-Loss Markov Random Fields"
8	Arora et al. [22]	2019	Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics	"A semi-Markov structured support vector machine model for high-precision named entity recognition"
9	Lay et al. [23]	2019	International Journal of Trend in Scientific Research and Development (ijtsrd)	"Myanmar Named Entity Recognition with Hidden Markov Model"
10	Drovo et al. [24]	2019	The 7th International Conference on Smart Computing & Communications (ICSCC)	"Named Entity Recognition in Bengali Text Using Merged Hidden Markov Model and Rule Base Approach"
11	Malik et al. [25]	2017	Pakistan Journal of Engineering and Applied Sciences	"Urdu named entity recognition system using hidden Markov model"
12	Leaman et al. [26]	2016	Bioinformatics	"TaggerOne: joint named entity recognition and normalization with semi-Markov Models"
Parts of Speech Tagging				
13	Azeraf et al. [27]	2020	arXiv preprint	"Hidden Markov Chains, Entropic Forward-Backward, and Part-Of-Speech Tagging"
14	Abdur Rohman et al. [28]	2019	"The 3rd International Conference on Informatics and Computational Sciences (ICICoS)"	"Twitter Storytelling Generator Using Latent Dirichlet Allocation and Hidden Markov Model POS-TAG (Part-of-Speech Tagging)"
15	Assunção et al. [29]	2019	"International Journal of Software Engineering and Knowledge Engineering"	"Language Independent POS-tagging Using Automatically Generated Markov Chains (S)"
16	Kadim et al. [30]	2018	"The International Arab Journal of Information Technology"	"Parallel HMM-based approach for Arabic part of speech tagging"
17	Afini et al. [31]	2017	"The 1st International Conference on Informatics and Computational Sciences (ICICoS)"	"Morphology analysis for Hidden Markov Model based Indonesian part-of-speech tagger"
18	Stratos et al. [32]	2016	"Transactions of the Association for Computational Linguistics"	"Unsupervised part-of-speech tagging with anchor hidden Markov models"

For each paper, we have focused on methods/techniques that employed, advantages and disadvantages. We discuss them in more detail in Section 5 and 6.

5. Literature Review of Applications of Markov Models in Natural Language Processing

Markov models are widely used in NLP in different domains. They can be used as individual algorithms or combined with different models, such as Hidden Markov Models can be used to train deep learning and neural networks models for different purposes such as discovering the language from speech or other applications [33]. In this paper, three subfields of NLP are presented in terms of using Markov models: Natural Language Generation, Named-Entity Recognition, and Parts of Speech Tagging.

5.1. Natural Language Generation

Natural Language Generation (NLG) generates text in several human languages, based on data produced by humans. Nowadays, NLG supports humans in writing weather reports, routine business letters, and NLP is widely employed for the automatic generation of questions and answering systems [34]. In general, NLG approaches use rules, instructions, and heuristics to create acceptable and stylized reactions and generate text without notable differences from natural human language. However, generalization and scalability are not easy tasks, as rules or methods implemented in a specific language are not appropriate for another language [35]. The text generator for a language's has certain characteristics that help it to be reliable: efficiency, fluency, diversity, and readability. Researchers work in different ways to solve NLG problems and achieve those characteristics. Figure 3 shows the categories of NLG techniques.

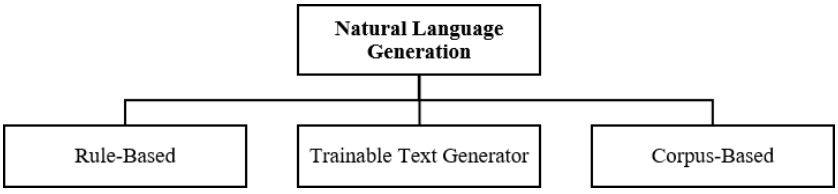

Fig.3. Natural language generation approaches [35].

Rule-Based

The rule-based [36] or template-based method is one of the effective ways widely used to generate text. Despite its effectiveness and robustness, the repetition of the same output makes the rule-based method rather tiresome. In addition, the method has some difficulties in scaling to multiple open-domain systems. Therefore, NLG techniques need to be easily scaled up to meet the above characteristics.

Trainable Text Generator

The trainable text generator [37] provides certain trainable models that help scalability and enable the model to adjust to additional domains or *reproduce* a particular style. However, this method has some restrictions: it needs a handcrafted generator to describe the decision options as well as statistical approaches that may help optimization. Therefore, any new domain-specific responses or output require manual additions based on predefined syntax.

Corpus-Based

Nowadays, corpus-based [38, 39] approaches, in which a huge amount of data has been generated by humans, are widely used and have become available on different platforms and in datasets. Corpus-based approaches use machine learning and statistical models to learn from data, generating a stochastic list of candidate words that usually come in a specific sequence according to what the model learns from data. Data generated by humans help systems create a more natural text and alike to human responses. Thus, it reduces the dependence on manual intervention and predefined rules.

Application of Markov Models in Natural Language Generation.

Markov models are widely employed in natural text generation. Zhang et al. [15] employed Metropolis-Hastings Sampling (MHA) for generating fluent adversarial samples. Recently, adversarial learning has become a widespread subject in deep learning. It aims to generate new examples by disturbing the samples and utilizing them to deceive deep neural networks, known as a victim model. Generated adversarial examples are inserted into the training samples to enhance the victim model's effectiveness and robustness. Metropolis-Hastings's sampling is a technique for generating examples that rely on Markov chain Monte Carlo (MCMC) to create desirable examples from a sequence of random samples based on probability distribution when direct sampling is challenging. Their proposed MHA was applied to Internet Movie Database (IMDB) and Stanford Natural Language Inference (SNLI) datasets. The MHA presents lower perplexity (PPL) that indicates rather similar sentences to the generated examples in the corpus used. The proposed method obtained notable results of 73% in terms of accuracy. On the other hand, the MHA still returned examples, even when the label had been changed - their study relied on volunteers for annotation or labelling the generated examples - leading to generating incomplete sentences that are not considered fluent from the human perspective. To solve this problem, rule-based measures or constraints, such as identifying the end of the sentence (EOS), should be implemented before returning examples.

Neural models are mathematical or statistical frameworks that combine artificial neural networks, fuzzy logic, and other AI tools. Neural models were commonly employed in NLP, and the accuracy of those models depends on the quality of data. Martínez Garcia et al. [16] combined Markov Chains and Word Embedding for creating new samples - text sentences- to expand the training data. Markov chains were used to discover the following word in a series of words according to the current state. According to their model, trained on a Spanish corpus, they used a Markov chain to generate sentences that considered a new sample would be utilized on the training set. They then implemented cosine similarity to filter the sentences to help select sentences similar to what they used in the dataset. Their proposed method can be used as an oversampling approach to solve imbalanced datasets in supervised learning. It is more effective than Synthetic Minority Oversampling Technique or SMOTE. SMOTE only works to produce identical duplications of the same samples according to the nearest neighbors as an example.

Text summarization is among the state-of-art methods in NLP, implementing machine learning and mathematical models to reduce the words of the text and select or generate important parts as a sequence of sentences to summarize the whole text. Researchers have introduced different techniques to achieve a summary of the document similar to what would be produced by a human reader. Gehrman et al. [17] used the Semi-Markov Conditional Random Field, which relies on the Markov chain, to generate document titles. To achieve this, they applied three steps: Selector, Compressor, and Ranker. The Selector works to select the important sentence in each paragraph and illustrates each word using two separate embedding approaches. They then calculated the probability using Long Short Term Memory networks (LSTM) to select words. In the Compressor stage, they used the Markov model to reduce the sentence length, by removing unnecessary words. The Ranker was used to score the sentences to be selected and generated a title. They used Sequence-to-Sequence (S2S) [40], one of the text summarization techniques that uses the alignment between target and source sequences. However, since the S2S technique requires learning the alignment and generating words, it normally performs less well with limited data. They also suggested using end-to-end models to create a title most like in manner to that used by humans.

Harrison et al. [18] presented the application of Markov Chain Monte Carlo (MCMC) for generating a story, in which a summary of movies was produced in the form of a sequence of events. The Markov chain used for event

representation extracts a specific pattern from a sentence, for example, {subject; verb; object; modifier}. Then they used recurrent neural networks (RNNs) to summarize events and direct the MCMC search toward creating stories that satisfy genre expectations. They used romance movie summaries taken from Wikipedia as a dataset to test their proposed solution. They achieved acceptance criteria of nearly 85%. This approach can generate texts of the expected genre, but because their model's output is an event, not raw text, the movie plots are not interpretable. Moreover, there are some limitations in defining clear and effective acceptance criteria. Their method is in a preliminary stage and presented a good start and it was hoped future work would introduce more improvement.

Markov models are widely applied to steganography for security and encryption purposes. Steganography is a process of hiding secret information within text, images, or videos. Markov chains are employed to generate natural sentences used to encode or hide bits within words. Yang et al. [19] introduced steganography text which relied on Markov chains, which generated English sentences for hiding secret bits within words using Huffman encoding. Their method was applied to three large datasets, Twitter, IMDB for movie reviews, and News. The datasets were used to train a model to create a large dictionary or probability table. The use of such large datasets helped them to achieve a robust solution and generated natural sentences such as "I will say this is the best part of the film." Markov chain was employed to select m (e.g., 8) words from the large dictionary as a candidate pool with higher probabilities to come after the starting keyword. For example, after "I" the candidate pool will be {have, am, will, was, would, bought, got, can}. The proposed solution relied on a conditional probability distribution for dynamic coding of each word, which means the same word may have various codings. On the other hand, the results achieved an accuracy of between 50% to 60%, which would still need to be enhanced.

Luo et al. [20] presented a solution for hiding information using Markov chains to generate a classic Chinese poem named Ci-poetry. A Markov model was used as a corpus-based method to learn from data, in which it obtained a probability matrix for transferring between states based on simulated Ci-poetry in the Chinese language. They applied the proposed method to the Quansongci dataset ("Ci-poetry from Complete Collection of Song Period Ci-poetry") that contains 21000 Ci poems, which was considered a small dataset that thus caused chain breaks.

5.2. Named Entity Recognition

Named entity recognition (NER) is an essential subfield of NLP which aims to discover entities' names (person, city, company) from text. Named entities extraction is required to improve tasks such as question answering, referring expressions or pronouns in a sentence to the same entity (coreference resolution), and discovering semantic relationships between entities in sentences (relation extraction). Handcrafted methods were employed to tackle this problem in the past, but, with the development of data mining, and machine learning methods availability of large datasets, we now have the opportunity to leverage the power of machine learning to solve it [41]. There are different approaches for NEG, as shown in Figure 4: Knowledge-based (lexicon-based), rule-based, and machine learning-based (Corpus-based).

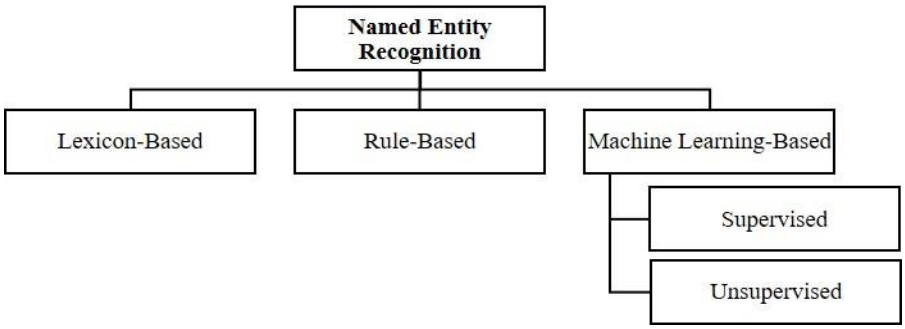

Fig.4. Named entity recognition approaches [42, 43].

Lexicon-Based Approaches

In lexicon-based approaches, there is no need for annotation tasks (labelling); the process relies on searching for entities in a predefined dictionary. This approach may work quickly, but there are some limitations in preparing a large dictionary, considered as a manual approach, and having low scalability for different domains [42].

Rule-Based Approaches

Rule-based techniques depend on regular expressions and predefined rules and are performed by searching for a specific predefined pattern or characteristics of the entities of interest [43]. For example, grammar rules and parts of speech may be used as a rule or pattern that needs to be discovered. The approach requires manual construction and is time-consuming.

Machine Learning-Based Approaches

A machine learning-based method can be supervised or unsupervised. In the supervised version, a model needs to

be trained on a dataset collected for a specific purpose or domain, and the model tries to predict new name entities. It reduces manual construction but still needs an annotation task for the training set and it is time-consuming to create corpora for different domains. It is useful when online learning is performed, in which it increases the corpus with new correct predictions. In the unsupervised version, a model trains on unlabeled data and tries to discover hidden patterns or characteristics of entities. Large datasets generated by humans, such as Wikipedia or movie reviews, are widely used to train a model. Deep learning and quantitative models such as Markov models are also used to extract features or calculate probabilities for words in a text to be candidates as an entity.

Application of Markov Models in Named Entity Recognition.

Bio-entity recognition plays a crucial role in medical information retrieval and expert systems, and question answering. Extracting names of biological entities like proteins and genes from scientific and medical documents is not a straightforward process because it depends on a context in which the entity can be either a gene or a protein. Miller et al. [21] presented a solution to overcome the drawbacks of methods like those of Huang et al., which tried to extract entities using context graphs that relied on indirect co-occurrence relationships among words These attempts did not achieve notable results in considering the semantics of these words. Therefore, Miller et al. introduced a probabilistic method to predict that something is a bio-entity using hinge-loss Markov random fields (HL-MRF) [44]. (HL-MRF) is a new type of graphical model developed to help scalable modeling of structured and rich data. Miller et al. combined associative information (e.g., two references that appear in the same abstract or could be a gene or protein) with embedding-based word semantics to classify bio-entities from documents. They used Probabilistic Soft Logic to represent an HLMRF: when the document contains a term “gene” it will be considered an observed variable; they then tried to determine a probability distribution over the unobserved variables. They obtained good results, 93.7% in terms of F score. This proposed method is very recent and may show better results if tested on different biomedical domains to support disambiguate references and discover entity based on a biomedical context.

Arora et al. [22] adopted BiLSTM-CNN "Bidirectional Long Short-Term Memory (BiLSTM)” and " for extracting named entity. They used the Markov model to reduce errors during training. They combined a semi-Markov method with a structured support vector machine to develop a custom loss function for increased precision in NER. The method also handles the trade-off or balance between and recall and precision by giving scores to various kinds of errors in inferencing the augmented loss throughout training. Employing Markov models for optimization is a good idea and can be used in different NLP tasks in addition to NER.

Lay et al. [23] introduced a supervised method to recognize a named entity in the Myanmar language. Hidden Markov Model was used to calculate start probability, transition probability and emission probability, according to a Myanmar labeled corpus. The HMM finds a state which comprises all the candidate named entities. It then predicts an unobserved entity based on the observed or candidate entity. Their method achieved good results, 95% and 97%, in terms of classification accuracy and F-score. However, it is a direct approach that relies on annotation tasks but still needs to be tested on different languages such as Arabic or Chinese to measure HMM's effectiveness. HMM helped to achieve precise prediction; therefore. it is preferred to test on different domains.

HMM was used with the same approach by Drovo et al. [24] for supervised classification in the Bengali language. However, they combined HMM with a rule-based method. For the rule-based approach, they used a regular expression in the form of grammar rules to discover an entity according to Regex matching. This means the text still needs to be manually annotated, and an unsupervised approach is preferred because of the limitations of covering all entities names in any language.

Malik et al. [25] presented HMM to predict NER for the Urdu language. They employed parts of speech to identify entities and found that the accuracy was improved from 66.71% to 71.70%. The same limitations apply as those mentioned for the previous two methods in [23, 24].

Leaman et al. [26] integrated NER with the normalization step for extracting names of diseases and chemical entities. Normalization was accomplished by assigning weights to all text parts for every NER category and the term in the lexicon. The Semi-Markov model was used to scale normalization vectors based on unit or segment length. This scaling enables balancing the normalization weights to make them independent of the total number of words in the lexicon name or the text's segment. This balancing needs data to be combined across the text segment, which is performed using the Semi-Markov model. Their proposed method tends to depend more on the lexicon when some names are unseen.

5.3. Parts of Speech Tagging

Parts of Speech (POS) tagging is an essential process in NLP. It is a procedure of tagging each word in a sentence with what part of speech it is. POS tags such as nouns, verbs, pronouns, prepositions, and adjectives assign meaning to a word and help the computer to understand sentences. There are different techniques and categories, as shown in Figure 5, to perform POS tagging [45, 46]: rule-based, stochastic or statistical, hybrid.

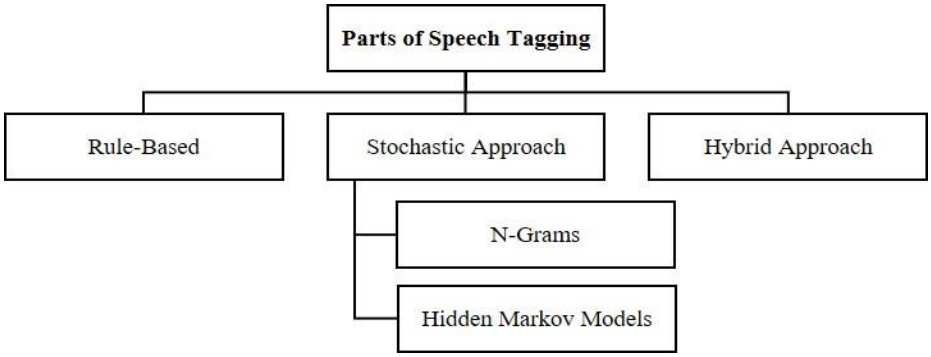

Fig.5. The different approaches of tagging parts of speech [45, 46].

Rule-Based approaches

The rule-based approach is the oldest method, which uses a dictionary or lexicon to match a word with a tag. It depends on manual construction and efforts, such as dataset annotations or lexicon creation. For example, a word that comes after "a" is a noun, a -> noun. A regular expression is also used to match text patterns for discovering a tag of a word. Knowledge-driven taggers, supported by human and built dictionaries, have achieved high accuracy results. On the other hand, it needs manual construction, and there are some limitations in using a defined number of rules [45].

Stochastic Approach

This method relies on computational models to automatically assign a tag to a word without the need to label data or create a dictionary. Statistics, frequency, and probability are used to couple a word with a tag. The tag is assigned based on the probability of a word appearing with a specific tag or the frequency of words that come after each other. It does not require a labeled lexicon but still needs a training corpus for calculating probabilities. In the stochastic method, n-grams or Hidden Markov Models (HMM) are widely used for POS tagging.

There are three types of n-grams: unigram (one word), bigram (two words), and trigram (three words). Table 7 shows the grams, using *I enjoyed the coffee* as an example.

Table 7. An example of n-grams in NLP

N-gram	Words/Sentences
Unigram	I enjoyed the coffee
Bigram	I enjoyed enjoyed the the coffee
Trigram	I enjoyed the enjoyed the coffee

This approach uses a statistical model to calculate the probability for grams and assigns a tag which corresponds as most likely with the determined grams. The probability of the unigram word is defined through the equation

$$P(g_i \mid d_i) = freq(d_i \mid g_i) \,/\, freq\,(d_i) \tag{2}$$

The probability of the bigram word is defined through the equation

$$P(g_i \mid d_i) = P(d_i \mid g_i)\,.\,P\,(g_i \mid g_{i-1}) \tag{3}$$

The probability of the trigram word is defined through the equation

$$P(g_i \mid d_i) = P(d_i \mid g_i)\,.\,P\,(g_i \mid g_{i-2}, g_{i-1}) \tag{4}$$

where g describes the sequence of tag, and w describes word sequence. $P(d_i \mid g_i)$ defines the likelihood of the current word provided the current tag, and $P\,(g_i \mid g_{i-1})$ the probability of the current tag provided the previous tag [45].

In POS tagging, HMM links each word in a text with a proper tag. In the POS, tags are hidden states, and a model tries to predict the tag according to observed words.

$$Find\ g_1^n such\ that\ \prod_{i=1}^{n} P(d_i \mid g_i)\,.P\,(g_i \mid g_{i-1}) \tag{5}$$

Hybrid Approach

The hybrid method combines rule-based and stochastic approaches. A model is trained using statistical techniques and a rule-based approach is also implemented to support accuracy and performance.

Application of Markov Models in Parts of Speech Tagging

Following this brief introduction to POS and the tagging technique, we will present recent studies that applied such techniques and their contributions. Azeraf et al. [27] combined Hidden Markov Model (HMM) with original Entropic Forward and Entropic Backward (EFB) possibilities. They first tried to prove HMM's effectiveness with classical EFB, which faces some difficulties manage arbitrary features such as suffixes or prefixes of different lengths, excluding with an independence rule. Therefore, the researchers developed a Maximum Entropy Markov Model (MEMM) to handle this problem. Their proposed method (HMM with EFB) proved to handle arbitrary features and showed better results compared to MEMM. Their method has been shown to be efficient and could be used as another option for Recurrent Neural Networks (RNN) to address sequential data with deep layers.

Abdul Rahman et al. [28] combined Latent Dirichlet Allocation (LDA) for Topic modelling and HMM for POS tagging, for a Twitter storytelling generator. They employed POS-TAG using HMM, which showed better accuracy results and faster performance, compared to different approaches in recent studies, such as the Maximum Entropy with Conditional Random Field (CRF), transformational-based method with CRF, etc. Thus, HMM is a robust method of POS tagging, which usually shows superiority over other techniques.

Assunção et al. [29] presented an independent POS-tagging solution using Markov chains. They tried to prove the power of Markov chains in POS tagging, compared to HMM, which was usually used for this task. Their proposed method is flexible and obtained good results, but it is a supervised method and needs manual annotation, which is a difficult and time-consuming task.

Kadim et al. [30] introduced parallel Hidden Markov Model for the Arabic language. They tried to solve the problem of deriving Arabic tagging concepts from English, which differs in its structures. They proposed two HMM taggers which work in parallel for performing tagging: the first tagger is the main one, and the second works as a reference for determining the text with low probability tags. They also used a linking matrix to calculate the probability of the two taggers and obtain the final results. Their methods achieved good results in terms of accuracy, 98.22% for the first tagger and 75.12% for the second tagger. On the other hand, they faced some limitations, such as the size of the Nemlar Arabic corpus, which needed to be enlarged. There were also some difficulties in the parallel implementation of HMM taggers. However, the idea of dividing the tagging process into more than modules working in parallel is an *exciting* idea, which could be used similarly to an ensemble classification or fusion feature selection.

Morphological Analysis in NLP is a process of discovering the smallest unit of the word that has a meaning and involves removing the prefix, suffix from words to get the root form. Afini et al. [31] combined morphological analysis with the Hidden Markov Model for the Indonesian Language, which deals with the out-of-vocabulary (OOV) problem. Out-of-vocabulary means the unknown words that never appear in the training set. Their method showed notable results with the unknown words and enhanced the accuracy of POS tagging in general. The idea of using morphological analysis help to reduce dependency on corpus size and lexicon construction.

Stratos et al. [32] introduced an unsupervised method using the Hidden Markov Model: it involved learning from unlabeled data to perform POS tags. The method is used to find anchor observations connected to potential POS tags across other languages, which means it can be used for another language without the need for an annotation task. This method aims to discover the correct sequence of hidden states (POS tags) when provided with a sequence of observation states, which are the words. The anchor state corresponds to "suppose there is at least one word that appears under each POS tag," which means the POS tag at least has one word. Such unsupervised approaches are reducing the cost of manual preparation for building a lexicon or annotating datasets. Nevertheless, speech tagging is still not an easy task, and it is challenging to evaluate the results.

6. Summary and Discussion

This section presents a summary of the methods discussed in the previous section together with evaluative comments highlighting the advantages and disadvantages of each study. Table 8 provides a summary of the discussed methods.

As we mentioned, the objective of this study investigates some current researches that introduced solutions for using Markov models in NLP. Table 8 summarized the methods and techniques they employed, contributions and advantages, and how the Markov models improve NLP, finally, it introduced the limitations and advantages of each study, to help interested researchers in distinguishing research gaps or even applying the benefits of employing Markov models in their research related to NLP.

Most researchers implemented a corpus-based approach in natural language generation because rule-based and trainable text generators are old, limited and require more effort. Additionally, their methods involved the supervised

approach, which relies on annotation tasks. There were some attempts to apply unsupervised solutions or reducing dependency on lexicons or labeled datasets. Markov chains were used more than HMM in text generation.

Table 8. Summary of discussed studies and their advantages and disadvantages

No	Authors	Year	Methods/ Techniques	Advantages/ Disadvantages
Natural Language Generation				
1	Zhang et al. [15]	2020	• Corpus-based • Markov chain • Metropolis-Hastings sampling	They employed the Markov chain to generate examples of rather similar sentences to those in the corpus they used. There are some incomplete sentences; therefore, they need to add rule-based techniques such as the end of the sentence (EOS) to hand this problem
2	Martínez Garcia et al. [16]	2020	• Corpus-based • Markov chain • Word Embedding	They combined Markov Chains and Word Embedding to create new samples. Their proposed method can be used as an oversampling approach to solve imbalanced datasets in supervised learning. It's better than Synthetic Minority Oversampling Technique or SMOTE, which only works to produce identical duplications of the same samples according to nearest neighbors as an example.
3	Gehrmann et al. [17]	2019	• Corpus-based • Semi-Markov Conditional Random Field • Text summarization • Sequence-to-Sequence	They used Semi-Markov Conditional Random Field which relies on the Markov chain, to generate a title for a document. Text summarization is not an easy task; they introduced an acceptable solution which may show significant results when some limitations are overcome. However, since the Sequence-to-Sequence (S2S) technique requires jointly learning the alignment and generating words, it is usually less effective with limited data. They suggest using end-to-end models to generate a title, similarly to the human manner.
4	Harrison et al. [18]	2017	• Corpus-based • Markov Chain Monte Carlo • Metropolis-Hastings sampling	They present Markov Chain Monte Carlo (MCMC) for generating a story, which was a summary of movies in the form of a sequence of events. There are some limitations in defining clear and convincing acceptance criteria, and the movie plots are not interpretable. Their method was in a preliminary stage and future work was expected to introduce further improvement.
5	Yang et al. [19]	2018	• Corpus-based • Markov chains • Steganography • Huffman encoding	They introduced steganography texts that relied on Markov chains which generate English sentences for hiding secret bits within words using Huffman encoding. The conditional probability distribution was used for dynamic coding of each word, which means the same word may have various codings. The accuracy achieved needed further enhancement
6	Luo et al. [20]	2016	• Corpus-based • Markov chains • Steganography • Huffman encoding	They presented a solution for hiding information using Markov chains to generate Ci-poetry. There were breaks in the Markov chains because the dataset was small.
Named-Entity Recognition				
7	Miller et al. [21]	2020	• Machine learning-based • HMM • Relational information • Embedding-basedword semantics	They introduced a probabilistic method to predict a bio-entity using hinge-loss Markov random fields. Combining associative information (like references appearing in the same abstract or could be a gene or protein) with embedding-based word semantics to classify a bio-entity from documents. They obtained good results, but their method needed to be tested on different biomedical domains to support disambiguating references and discover entities based on a biomedical context. Specific-domain ENR still relies on annotated datasets, The present authors hope there are some contributions to unsupervised methods.
8	Arora et al. [22]	2019	• Machine learning-based • HMM • BiLSTM-CNN. • Markov models as optimizers	They used the Markov model to reduce errors during training. They combined semi-Markov with a structured support vector machine to develop a custom-loss function for increased precision in NER. Implementing Markov models for optimization appears to be a promising idea that can also be used in different NLP tasks, not only NER.
9	Lay et al. [23]	2019	• Machine learning-based • HMM	They introduced a supervised method to recognize a named entity in the Myanmar language, using HMM. Their method achieved good results, but it is a direct approach that relies on annotation tasks and still needs to be tested on different languages such as Arabic or Chinese to measure the effectiveness of HMM. HMM helped in precise prediction, so could be tested on different domains.
10	Drovo et al. [24]	2019	• Machine learning-based • Rule-based • HMM	They combined HMM as a machine learning technique with a rule-based technique. For the rule-based approach, they used a regular expression in the form of grammar rules to discover an entity according to Regex matching. This still needed to be manually annotated, and an unsupervised approach would be preferable, because of the limitations of covering all entity names in any language.
11	Malik et al. [25]	2017	• Machine learning-based • HMM • Parts of speech as optimizer	They presented a method using HMM to predict NER for the Urdu language. Employing parts of speech (POS) helped in identifying entities and improved accuracy. The same limitations applied as those mentioned in [23, 24].
12	Leaman et al. [26]	2016	• Machine learning-based • Lexicon-based • Semi-Markov • Normalization	They integrated NER with the normalization step for extracting names for diseases and chemical entities. Normalization was accomplished by assigning weights to all text parts for every NER category and a term in the lexicon. Their proposed method tends to depend more on the lexicon when some names are unseen.

Parts of Speech Tagging				
13	Azeraf et al. [27]	2020	• Stochastic approach • HMM • Entropic Backward Forward and Entropic Backward (EFB)	They combined HMM with original Entropic Backward and Entropic Forward (EFB) probabilities. They tried to establish HMM's effectiveness with classical EFB. Their method showed efficiency and could be used as another option for RNN to address sequential data through deep layers.
14	Abdur Rohman et al. [28]	2019	• Stochastic approach • HMM	They combined Latent Dirichlet Allocation (LDA) for Topic modeling and HMM for POS tagging, for a Twitter storytelling generator. They employed POS-TAG using HMM, which showed better accuracy results and faster performance, compared to different approaches in recent studies, such as the Maximum Entropy with Conditional Random Field (CRF), transformational-based method with CRF, etc.
15	Assunção et al. [29]	2019	• Stochastic approach • Markov chains	They tried to prove the power of Markov chains in POS tagging, where HMM was usually used for this task. Their proposed method is flexible and obtained good results, but is a supervised method and needs manual annotation, which a difficult and time-consuming task.
16	Kadim et al. [30]	2018	• Stochastic approach • HMM • Parallel Processing	They introduced parallel Hidden Markov Models for the Arabic language. They tried to solve the problem of deriving Arabic tagging concepts from English, which differs in structures. The idea of dividing the tagging process into more than one module working in parallel is an innovative idea and could be used similarly to an ensemble classification or fusion feature selection.
17	Afini et al. [31]	2017	• Stochastic approach • HMM • Morphological analysis	They combined morphological analysis with HMM for the Indonesian language, to deal with the out-of-vocabulary (OOV) problem. The idea of using morphological analysis helps to reduce dependency on corpus size and lexicon construction.
18	Stratos et al. [32]	2016	• Stochastic approach • Unsupervised • HMM	They presented an unsupervised method using HMM; this involved learning from unlabeled data to perform POS tags. Such unsupervised approaches reduce the cost of manual preparation for building a lexicon or annotating datasets, But the task is still not easy and the results are difficult to evaluate.

In named-entity recognition, the machine learning-based approach was widely used. In this method, a model trained on labeled data, and entities are predicted. There was a tendency to use unsupervised methods in the context of studies in the English language and. in the future, quantitative models and deep learning will be used to help discover the relations between words to recognize entities. Studies in other languages relied on labeled data and performing direct classification. Recognizing entities in specific domains, such as biology, relied on a lexicon because in such cases the recognition is subject to the context in which the same word may has two categories. The Hidden Markov Model was used more frequently than Markov chains in NER.

In parts of speech tagging, the stochastic approach was implemented more than the other methods. This method was still supervised and relied on lexicons in different languages, except in the case of studies in English, which tried to reduce this dependency. Morphological analysis and Parallel Processing are noteworthy ideas in POS tagging, which may evolve in the future. The Hidden Markov Model was utilized more than Markov chains, although one study [29] attempted to prove the power of Markov chains in POS tagging.

It was determining the limitations of other studies to help interested researchers to find research gaps or even employ the benefits of using Markov models in their research proposals according to the information that has been summarized in this study. Undoubtedly, this value will be followed by successes in an academic aspect, such as creating motivations for writing more scientific papers that serve the NLP area by using Markov models.

7. Conclusion

This study was focused on applying Markov models in NLP which was discussed some of the most recent studies that focused on methods/techniques, contributions/advantages, and limitations/disadvantages of utilizing Markov models in NLP. It was aimed to motivate the researchers to find some gaps such as to move to unsupervised methods in NLP for reducing annotation and manual tasks.

A Markov model is a method for calculating the probability of anything happening in the future based on the probabilities we now have. These models are commonly used in the business sector, notably for market share analysis, meteorology, education for estimating future student enrolment, and manufacturing for evaluating the chance of machinery breaking at some time in the future. Because natural language is a probabilistic language that relies on a sequence of words to convey meaning in context, stochastic models such as Markov models are appropriate.

Natural Language Processing (NLP) is similar to many disciplines that rely on statistical modeling. Statistical NLP draws inferences from statistics to be employed in NLP. This involves acquiring data that have been generated through an unknown probability distribution and drawing inferences from them. One of the most effective means of undertaking NLP is the use of Markov models in machine learning. In general, NLP processes like Natural Language Generation, Named-Entity Recognition, and Parts of Speech Tagging use different techniques to perform NLP tasks. These techniques fall into three main categories. The rule-based type is the oldest approach, which depends on predefined instructions or patterns to help a model match a text to these criteria to extract terms or generate a sequence of words in

sentence form. The lexicon-based approach relies on a dictionary, and a model will search within this dictionary. Morphological analysis helps to reduce dependency on lexicons for each word, by extracting the root form of words. Both methods are time-consuming and need manual construction. The corpus-based (machine learning-based, learning-based) approach relies on training a model on a dataset that is labeled (supervised) or unlabeled (unsupervised) to predict/discover the desired output.

Most NLP studies reviewed in this paper implemented supervised methods with enhancement using statistical models or deep learning to reduce the dependency on annotation tasks. There were some attempts to apply unsupervised solutions for reducing dependency on a lexicon or labeled datasets in natural language generation. Markov chains were used more frequently than HMM in text generation. Markov chains are widely used in natural language generation. In named-entity recognition, there was a tendency to use unsupervised techniques in studies in the English language medium, and to use quantitative models and deep learning to find the relations between words to recognize entities. In studies involving the other languages, the approaches depended on using labeled data and conducting direct classification. Recognizing entities in specific domains, such as biology, is still dependent on a lexicon, because it relies on a context. The Hidden Markov Model was utilized more than Markov chains in both NER and POS tagging. Morphological analysis and parallel processing are useful ideas in POS tagging and may have more potential for future development.

Finally, this study has discussed some of the current researches that use Markov models in NLP. It aimed to discover limitations and gaps that may help researchers to improve NLP tasks and applications using Markov models or combining them with machine learning, deep learning to get high performance and more accurate results.

References

[1] H. M. Hapke, H. Lane, and C. Howard, “Natural language processing in action.” Manning, 2019.
[2] C. Manning and H. Schutze, Foundations of statistical natural language processing. MIT press, 1999.
[3] M. A. M. Bhuiyan, “Predicting stochastic volatility for extreme fluctuations in high frequency time series,” 2020.
[4] B. Render and R. M. Stair Jr, Quantitative Analysis for Management, 12e. Pearson Education India, 2016.
[5] P. N. Reddy and G. Acharyulu, Marketing research. Excel Books India, 2009.
[6] D. JURAFSKY and H. M. JAMES, “Speech and language processing. 3rd edn.,” Online: https://web. stanford. edu/~ jurafsky/slp3, 2019.
[7] D. S. Myers, L. Wallin, and P. Wikström, “An introduction to Markov chains and their applications within finance.” Mathematical Sciences-Chalmers University of Technology and University of …, 2017.
[8] L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite state Markov chains,” Ann. Math. Stat., vol. 37, no. 6, pp. 1554–1563, 1966.
[9] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” Proc. IEEE, vol. 77, no. 2, pp. 257–286, 1989.
[10] M. PIETRZYKOWSKI and W. SAŁABUN, “Applications of Hidden Markov Model: state-of-the-art,” Int. J. Comput. Technol. Appl., vol. 5, no. 4, pp. 1384–1391, 2014.
[11] G. D. Forney, “The viterbi algorithm,” Proc. IEEE, vol. 61, no. 3, pp. 268–278, 1973.
[12] V. Irechukwu, “Overview of The Hidden Markov Model (HMM)— What it can do for you in Machine Learning,” medium, 2018. https://medium.com/@victor.irechukwu/overview-of-the-hidden-markov-model-hmm-what-it-can-do-for-you-in-machine-learning-83b3003297b9 (accessed Oct. 17, 2020).
[13] V. Sharma, A. Panchal, and V. Y. Rane, “AN ANALYSIS ON CURRENT RESEARCH TRENDS AND APPLICATIONS OF NATURAL LANGUAGE PROCESSING,” Adv. Innov. Res., p. 63, 2020.
[14] A. Chopra, A. Prashar, and C. Sain, “Natural language processing,” Int. J. Technol. Enhanc. Emerg. Eng. Res., vol. 1, no. 4, pp. 131–134, 2013.
[15] H. Zhang, H. Zhou, N. Miao, and L. Li, “Generating fluent adversarial examples for natural languages,” arXiv Prepr. arXiv2007.06174, 2020.
[16] E. Martínez Garcia, A. Nogales, J. Morales Escudero, and Á. J. Garcia-Tejedor, “A light method for data generation: a combination of Markov Chains and Word Embeddings,” 2020.
[17] S. Gehrmann, S. Layne, and F. Dernoncourt, “Improving human text comprehension through semi-Markov CRF-based neural section title generation,” arXiv Prepr. arXiv1904.07142, 2019.
[18] B. Harrison, C. Purdy, and M. O. Riedl, “Toward Automated Story Generation with Markov Chain Monte Carlo Methods and Deep Neural Networks.,” 2017.
[19] Z. Yang, S. Jin, Y. Huang, Y. Zhang, and H. Li, “Automatically generate steganographic text based on Markov model and Huffman coding,” arXiv Prepr. arXiv1811.04720, 2018.
[20] Y. Luo, Y. Huang, F. Li, and C. Chang, “Text Steganography Based on Ci-poetry Generation Using Markov Chain Model.,” TIIS, vol. 10, no. 9, pp. 4568–4584, 2016.
[21] A. Miller, N. Markenzon, V. Embar, and L. Getoor, “Collective Bio-Entity Recognition in Scientific Documents using Hinge-Loss Markov Random Fields,” 2020.
[22] R. Arora, C.-T. Tsai, K. Tsereteli, P. Kambadur, and Y. Yang, “A semi-markov structured support vector machine model for high-precision named entity recognition,” in Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 5862–5866.
[23] K. K. Lay and A. Cho, “Myanmar Named Entity Recognition with Hidden Markov Model,” 2019.
[24] M. D. Drovo, M. Chowdhury, S. I. Uday, and A. K. Das, “Named Entity Recognition in Bengali Text Using Merged Hidden Markov Model and Rule Base Approach,” in 2019 7th International Conference on Smart Computing & Communications

(ICSCC), 2019, pp. 1–5.
[25] M. K. Malik and S. M. Sarwar, “Urdu named entity recognition system using hidden Markov model,” Pakistan J. Eng. Appl. Sci., 2017.
[26] R. Leaman and Z. Lu, “TaggerOne: joint named entity recognition and normalization with semi-Markov Models,” Bioinformatics, vol. 32, no. 18, pp. 2839–2846, 2016.
[27] E. Azeraf, E. Monfrini, E. Vignon, and W. Pieczynski, “Hidden Markov Chains, Entropic Forward-Backward, and Part-Of-Speech Tagging,” arXiv Prepr. arXiv2005.10629, 2020.
[28] Y. A. Rohman and R. Kusumaningrum, “Twitter Storytelling Generator Using Latent Dirichlet Allocation and Hidden Markov Model POS-TAG (Part-of-Speech Tagging),” in 2019 3rd International Conference on Informatics and Computational Sciences (ICICoS), 2019, pp. 1–6.
[29] J. Assunção, P. Fernandes, and L. Lopes, “Language Independent POS-tagging Using Automatically Generated Markov Chains (S).,” in SEKE, 2019, pp. 513–666.
[30] A. Kadim and A. Lazrek, “Parallel HMM-based approach for arabic part of speech tagging.,” Int. Arab J. Inf. Technol., vol. 15, no. 2, pp. 341–351, 2018.
[31] U. Afini and C. Supriyanto, “Morphology analysis for Hidden Markov Model based Indonesian part-of-speech tagger,” in 2017 1st International Conference on Informatics and Computational Sciences (ICICoS), 2017, pp. 237–240.
[32] K. Stratos, M. Collins, and D. Hsu, “Unsupervised part-of-speech tagging with anchor hidden markov models,” Trans. Assoc. Comput. Linguist., vol. 4, pp. 245–257, 2016.
[33] Valentin Gazeau, Cihan Varol, "Automatic Spoken Language Recognition with Neural Networks", International Journal of Information Technology and Computer Science(IJITCS), Vol.10, No.8, pp.11-17, 2018. DOI: 10.5815/ijitcs.2018.08.02
[34] M. Saad, S. Aslam, W. Yousaf, M. Sehnan, S. Anwar, and D. Rehman, “Student Testing and Monitoring System (Stms) Using Nlp.,” Int. J. Mod. Educ. Comput. Sci., vol. 11, no. 9, 2019.
[35] T.-H. Wen, M. Gasic, N. Mrksic, P.-H. Su, D. Vandyke, and S. Young, “Semantically conditioned lstm-based natural language generation for spoken dialogue systems,” arXiv Prepr. arXiv1508.01745, 2015.
[36] D. Mirkovic and L. Cavedon, “Dialogue management using scripts.” Google Patents, Oct. 18, 2011.
[37] F. Mairesse and M. A. Walker, “Controlling user perceptions of linguistic style: Trainable generation of personality traits,” Comput. Linguist., vol. 37, no. 3, pp. 455–488, 2011.
[38] F. Mairesse and S. Young, “Stochastic language generation in dialogue using factored language models,” Comput. Linguist., vol. 40, no. 4, pp. 763–799, 2014.
[39] T.-H. Wen et al., “Stochastic language generation in dialogue using recurrent neural networks with convolutional sentence reranking,” arXiv Prepr. arXiv1508.01755, 2015.
[40] I. Sutskever, O. Vinyals, and Q. V Le, “Sequence to sequence learning with neural networks,” in Advances in neural information processing systems, 2014, pp. 3104–3112.
[41] Nabil Ibtehaz, Abdus Satter, "A Partial String Matching Approach for Named Entity Recognition in Unstructured Bengali Data", International Journal of Modern Education and Computer Science(IJMECS), Vol.10, No.1, pp. 36-45, 2018.DOI: 10.5815/ijmecs.2018.01.04
[42] V. Yadav and S. Bethard, “A survey on recent advances in named entity recognition from deep learning models,” arXiv Prepr. arXiv1910.11470, 2019.
[43] T. Eftimov, B. Koroušić Seljak, and P. Korošec, “A rule-based named-entity recognition method for knowledge extraction of evidence-based dietary recommendations,” PLoS One, vol. 12, no. 6, p. e0179488, 2017.
[44] S. H. Bach, M. Broecheler, B. Huang, and L. Getoor, “Hinge-loss markov random fields and probabilistic soft logic,” J. Mach. Learn. Res., vol. 18, no. 1, pp. 3846–3912, 2017.
[45] S. G. Kanakaraddi and S. S. Nandyal, “Survey on parts of speech tagger techniques,” in 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), pp. 1–6, 2018.
[46] D. Kumawat and V. Jain, “POS tagging approaches: A comparison,” Int. J. Comput. Appl., vol. 118, no. 6, 2015.

Authors’ Profiles

Talal Almutiri is a PhD student at the Information Systems Department, Faculty of Computing and Information Technology, King Abdulaziz University, KSA. He received his BSc and MSc in Information Systems from Taibah University. His research interests are machine learning, natural language processing (NLP), and bioinformatics, and he has published different papers in his interest domains.

Farrukh Nadeem is a gold medalist in BSc. and completed MSc. Computer Science from the University of Punjab, Pakistan. He completed his PhD. with distinction in Computer Science in 2009 from the University of Innsbruck, Austria. He is an associate professor at the Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah. He has held several distinctions and awards during his educational career. He has been involved in several Austrian research projects and is working on a couple of Saudi research and development projects. He has gained professional training on Cloud Computing and High-Performance Computing. He has set up a “Grid Computing Infrastructure” at the Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah. He is a member of the program committees of several conferences and editorial board member of Journal of Modern

Education and Computer Science. Farrukh has authored more than 29 conference and journal research papers, including four book chapters. He has been awarded President (King Abdulaziz University) Certificate of Appreciation and cash award for one of his journal publications. His main research interests include performance modeling and prediction, the Internet of Things, and smart healthcare.

How to cite this paper: Talal Almutiri, Farrukh Nadeem, "Markov Models Applications in Natural Language Processing: A Survey", International Journal of Information Technology and Computer Science(IJITCS), Vol.14, No.2, pp.1-16, 2022. DOI: 10.5815/ijitcs.2022.02.01