# Hidden Markov Models for Sentiment Analysis in Social Media

Isidoros Perikos
*Computer Engineering and Informatics Department*
*University of Patras*
*Computer Technology Institute &Press "Diophantus"*
Patras, Greece
perikos@ceid.upatras.gr

Spyridon Kardakis
*Computer Engineering and Informatics Department*
*University of Patras*
Patras, Greece
kardakis@ceid.upatras.gr

Michael Paraskevas
*Department of Electrical & Computer Engineering*
*University of Peloponnese and Computer Technology Institute & Press "Diophantus"*
Patras, Greece
mparask@cti.gr

Ioannis Hatzilygeroudis
*Computer Engineering and Informatics Department*
*University of Patras*
Patras, Greece
ihatz@ceid.upatras.gr

***Abstract*—User-generated content in social media contain very meaningful information about public sentiment and stance and require automated methods to analyze them extract knowledge from them. The recognition of sentiments, emotions and attitudes in textual data is necessary for understanding public stance and is highly desired for various services and procedures. In this work, we examine the performance of Hidden Markov Models (HMM) in the recognition of sentiments and opinions in text. Hidden Markov Models constitute a quite suitable approach given the special characteristics of the textual data. In particular, they can utilize the sequential nature of textual data, a piece of information that traditional machine learning approaches fail to fully take into account. We performed several experiments in order to assess the performance of different HMM-based methods under various training parameters and architectures. The evaluation results indicate that Hidden Markov Models achieve superior performance compared to traditional machine learning algorithms and highlight that they are scalable and accurate in analyzing user generated content and in specifying opinions and attitudes.**

***Keywords—Sentiment analysis, Emotion recognition, Hidden Markov Models, Big Data***

## I. Introduction

Textual Sentiment Analysis has attracted increased interest from researchers in recent years due to the rise of social media and the Internet. Social networks, web-based applications and services give users the ability to express their opinions, attitudes and sentiments in the form of text and share it with millions of people around the globe [13]. Every day, a vast amount of textual content is posted in news portals, social networks and web 2.0 applications that is rich in opinions and attitudes, thus giving us access to an enormous volume of data. These data contain very meaningful information and automated methods are needed in order to analyze and extract knowledge from them [16]. Sentiment analysis methods aim to people's opinions, emotions and attitudes towards entities such as products, services, events etc. The goal is to extract knowledge about the attitude, the emotional state, and the subjective opinions of users [11].

The big quantity of opinion-heavy texts on the Internet shows that effective sentiment analysis is needed. Analyzing, recognizing and understanding sentiment and emotions in text is vital in order to grasp public opinion on various topics and events [19] [12]. Overall, sentiment analysis focuses on processing user-generated content such as products, services reviews, forums, blogs. As a result of this, it's apparent that businesses would benefit from automatically processing and understanding their users' content, whether that is a review or an opinion on an event [23]. The accurate recognition of opinions and sentiments in textual data is a very challenging task on its own and when it comes to the analysis of user-generated data in social media, things can get even more challenging [2][14].

In this paper, we examine the performance of Hidden Markov Models (HMMs) on sentiment analysis tasks under various conditions. Hidden Markov Models aim to utilize the sequential nature of textual data, which provides meaningful information that other methods fail to utilize. Given the special characteristics of the task of sentiment analysis and textual emotion recognition, utilizing Hidden Markov Models is a suitable and potent approach in order to increase the accuracy of classification since HMMs can take into account a sequence of data. Indeed, this constitutes a very indicative feature of textual information that traditional machine learning methods neglect to consider. To the best of our knowledge, such a wide variety of Hidden Markov Models has not been thoroughly studied for text classification tasks. We perform a variety of experiments on high-order Hidden Markov Models and expand on their usage for text classification and its challenges. Overall, we evaluate our models on public sentiment analysis datasets. The results show that the Hidden Markov Models achieve high accuracy on quite small datasets and outperform machine learning approaches that do not utilize the sequential nature of data. Hidden Markov Models achieve superior performance even without using word semantics. The evaluation results are quite interesting, showcasing sound improvement compared to other classification approaches, which indicates that the Hidden Markov Models are scalable as well as accurate in analyzing user-generated content and in specifying users' opinions and attitudes.

The rest of this paper is structured as follows: Section II reviews related work on the utilization of Hidden Markov Models in sentiment analysis domain. Section III demonstrates our methodology as well as the main background topics on Hidden Markov Models. Section IV presents the experimental

978-1-7281-0886-5/19/$31.00 ©2019 IEEE
BCD 2019, May 29-31, 2019, Honolulu, Hawaii

Authorized licensed use limited to: The Claremont Colleges Library. Downloaded on March 07,2023 at 21:12:56 UTC from IEEE Xplore. Restrictions apply.

study and analyses the results collected. Finally, Section V concludes the paper and provide main directions for future work.

## II. Related Work

Over the last years, the domain of sentiment analysis and by extension emotion recognition in social media has attracted newly found interest. There exists a huge research interest and several works that attempt to perform emotion detection as well as study the way people express emotions in the web and in social media [4] [6]. A detailed and complete overview of approaches can be found in [7] [11] with a plethora of works focusing on emotion expression in news, web blogs and social networks.

Hidden Markov Models have been recently examined by the research community and achieved promising results. In the work presented in [15], the authors present a weighted high-order HMM for a sentiment analysis task and in particular emotion recognition. The main focus of the authors is to combine HMMs of different orders including the common first-order HMM. The advantage of a high-order HMM is that it takes into consideration a number of previous transitions/events instead of only the directly previous one. The architecture of the system consists of: (1) a MaxEnt classifier to recognize the emotion of words, (2) recognition of some of the sentences (known states) (3) the proposed weighted high-order HMM which further recognizes the remaining sentences using modified (part of known states) Viterbi algorithm. An advantage of this approach is that it does not require word labels since a classifier is used to infer them. Furthermore, a representation of compound emotions is introduced, named emotion code. HMMs up to an order of seven are evaluated with the fourth-order performing the best, thus concluding that the emotions of a sentence are clearly affected by the direct previous three or four consecutive sentences.

In [17], authors propose the use of a HMM for sentiment analysis of customer reviews. The task at hand is to automatically analyze whether the comment given by a customer is positive or negative (polarity), since manually going through all customers' review to decide on the performance of the product would not be realistic. The HMM works on Part-of-Speech tagged sentences and will automatically extract the customer opinions present in the review comments on various product features. The model is evaluated on a multi-domain Amazon review dataset. It shows promising results, achieving an accuracy as high as 95.3% and an f-measure of 97.1%.

In the work presented in [9] authors propose a system that includes HMMs and an Ensemble for sentiment analysis. The task at hand is polarity classification (positive/negative). The main idea is to transform the input data/space into clusters and then employ HMMs for the classification, using those clusters as hidden states. The training instances are split according to their class label in order to then train one HMM on each subset. Then, for the evaluation phase, each instance "runs" through every model in order to compute a sentiment orientation. It is worth noting that if clustering was not performed and the original bag-of-words space was used, the calculated transition probability from word to word would be so small that the observation probability of a sentence pattern would be quite close to zero and practically useless. Finally, the Ensemble method used to convert weak HMMs into stronger models is Boosting. This approach is evaluated against state-of-the-art methods, including neural networks, on benchmark datasets and outperforms them with an accuracy as high as 87% on the Customer reviews Dataset and 98.1% on the Movie Review Subjectivity Dataset respectively.

In the work presented in [10], authors propose a modified version of a HMM for emotion classification in microblogs. The HMM operates on Ekman's six basic emotions and utilizes metrics such as mutual information and tf-idf of the text as observed variables. Those metrics are given as an input feature vector to an algorithm that calculates a probability. Then, the parameters and the hidden states of the proposed model are calculated with the help of the particle swarm optimization algorithm – a self-adaptive approach. Six HMMs are constructed, one for each of the six emotions and each new tweet is evaluated on all of them. Furthermore, since manually labeling large-scale datasets is close to impossible, the authors employ an entropy metric to decide whether an unlabeled tweet shall be contained in the training dataset after being assigned an emotion by the HMM. The proposed approach is evaluated against Naïve Bayes and SVMs on a dataset consisting of Chinese tweets. It outperforms them on almost all emotion categories with an f1-score as high as 85%.

In [8], the authors propose a HMM for classification of biomedical text documents. First, preprocessing is performed on the textual data, and words in each document are sorted by their relevance with the goal of extracting the, for example, top four relevant words. Then, the main idea is simple: split the data depending on their class label on two subsets and train two HMMs where the hidden states represent the previously mentioned most relevant words. The most probable observations for the first state are the most relevant words in the corpus, for the second state they are the words holding the second level of relevance in the corpus, and so on. An adaptation mechanism is also introduced, making the model able to learn from new documents, once trained.

In the literature, the majority of HMMs are trained using either the Baum-Welch algorithm or by simply counting transition and observation occurrences between states for labeled data (supervised) scenarios.

## III. Methodology

In this section, we will introduce our Hidden Markov Model-based approach for a sentiment analysis task and in particular for sentiment polarity classification. We train a single supervised HMM for each class label $y$ of the dataset, using a set of observations and their labels, which can refer to either sentences, phrases or words. For example, a dataset includes $n$ texts/documents, each consisting of a set of sentences $S$, forming a sequential vector: $d = [s_1, s_2, \dots, s_n]$. Every sentence is annotated with a label $x$ – commonly 'positive' / 'negative' / 'neutral'.

### A. Hidden Markov Models

A Hidden Markov Model is a statistical Markov model in which the system being modeled is assumed to be a Markov



Authorized licensed use limited to: The Claremont Colleges Library. Downloaded on March 07,2023 at 21:12:56 UTC from IEEE Xplore. Restrictions apply.

process with unobserved (i.e. hidden) states. It is a structured probabilistic model that describes a probability distribution over a set of possible sequences; they can be thought of as a general mixture model plus a transition matrix. Starting from some initial states with the initial probability, a sequence of states is generated by moving from one state to another according to the state-transition probabilities until a final state is reached, creating an observable sequence of symbols as each state emits a symbol when it is visited. The hidden states form a Markov chain with transition probabilities that satisfies the Markov property of depending only on the previous event (memorylessness). This means that in order to predict the next observation in a sequence, the distribution of predictions will depend only on the value of the immediately preceding observation and will be independent of all earlier observations. The latent variables – called hidden states – are discrete, while the observations can be either discrete or continuous with a variety of different distributions being able to model them. In general, a specific left-to-right architecture is preferred.

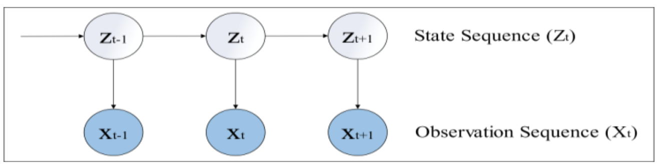


Fig. 1. The basic structure of Hidden Markov Model

In a HMM, the state is not directly visible to the observer, but the observations (outputs) are in fact visible. The "hidden" keyword refers to the states and not to the parameters of the model or the observation sequence and does not indicate that the parameters are unknown. A strength of HMMs is that they can model variable length sequences.

A Hidden Markov Model (HMM) is represented by a tuple $\theta = (S, V, \pi, A, B)$ which has the following parameters:

$S = \{S_1, S_2, \ldots, S_n)$ is a set of elements which are called states.

$V = \{V_1, V_2, \ldots, V_m)$, is a set of observations.

$\pi = \pi_1, \pi_2, \ldots, \pi_n$ with $\pi_i = P(s_i)$, is a vector of initial probabilities referring to the initial state distribution, for which, $0 \leq \pi_\iota \leq 1$, and $\sum_i \pi_i = 1$.

$a_{ij} = P(s_i|s_j)$, is the matrix of transition probabilities which is of size n×n and represent the transition probability from hidden state $i$ to hidden state $j$.

$b_i(v_m) = P(v_i|s_i)$, is the matrix of observation probabilities.

A HMM defines a probability distribution on a sequence of observations – e.g. sequence of words for an NLP task – of length $m$. It assumes that each observation is emitted by a corresponding hidden state variable $S_i$, the value of which determines the observation probabilities.

A distinct advantage of HMM is their ability to exhibit some degree of invariance to local warping of the time axis. Variations can be accommodated by the Hidden Markov Model through changes in the number of transitions to the same state versus the number of transitions to the successive state [3].

### B. *Training Procedure*

The most common algorithm to estimate the unknown parameters of a Hidden Markov Model is the Baum-Welch (BW) algorithm which is also known as the forward-backward algorithm. It is an expectation maximization algorithm and attempts to find the maximum likelihood estimate of the parameters of a HMM given a set of observed feature vectors. Observations are given weights which can then be used in the weighted Maximum likelihood estimation. As far as the forward-backward algorithm is concerned, it makes use of the principle of dynamic programming to efficiently compute the values that are required to obtain the posterior marginal distributions in two passes. The first pass goes forward in time while the second goes backward in time; hence the name forward–backward algorithm. It is a hill-climbing algorithm and can get stuck in local maxima, thus does not guarantee a global maximum. There exist several variants of the algorithm, all of which lead to exact marginals.

Given two models $\lambda$ and $\lambda'$, and a function Q, the idea is to iteratively improve the model parameters with the aid of Q. Start from some initial model $\lambda$ and find the Baum-Welch re-estimate, $\lambda'$, by maximizing the Q-function. Now take $\lambda'$ to be the new initial model and repeat the process. Eventually one of the following will happen: either the probability based on $\lambda$ will satisfy $P_{\lambda'} > P_\lambda$ or $\lambda$ is a critical point of $P_\lambda$. Since $P_\lambda$ has a finite number of critical points, iterating to convergence from initial models obtains a good estimate of the maximum likelihood model $\lambda'$. The overall process is as follows:

(1) Start with initial probability estimates referring to a model $\lambda$.

(2) Compute expectations of how often each transition/emission is used, which refer to model parameters of $\lambda$.

(3) Apply changes to the model in order to maximize paths.

(4) Re-estimate the probabilities iteratively until convergence.

If we were to describe a hidden Markov chain by $\theta = (\pi, A, B)$ then the Baum-Welch algorithm finds a local maximum for $\theta^* = argmax_\theta P(Y|\theta)$. For the forward procedure, let $\alpha_i(t) = P(o_1, o_2, \ldots, o_t, x_t = i|\theta)$, the probability of seeing $o_1, o_2, \ldots, o_t$ and being in state $i$ at time $t$. This is found iteratively:

$$a_i(1) = a[x_0, x_i]b[x_i, o_1]$$

$$a_j(t+1) = \sum_{i=1}^{N} a_i(t)a[x_i, x_j]b[x_j, o_{t+1}]$$

Then, for the backward procedure, let

$$\beta_i(t) = P(o_{t+1}, o_{t+2}, \ldots, o_T \mid x_t = i, \mu)$$

be the probability of the ending partial sentence $o_{t+1}, o_{t+2}, \ldots, o_T$ given starting state $i$ at time $t$. This is calculated by:

$$\beta_i(T) = 1$$

$$\beta_i(t) = \sum_{j=1}^{N} a[x_i, x_j]b[x_j, o_{t+1}]\beta_j(t+1)$$



Authorized licensed use limited to: The Claremont Colleges Library. Downloaded on March 07,2023 at 21:12:56 UTC from IEEE Xplore. Restrictions apply.

Now, for the updating phase, we can utilize the Bayes theorem. The probability of a state $i$ at time $t$ is:

$$\gamma_i(t) = P(x_t = i|O,\mu) = \frac{P(x_t = i, O|\mu)}{P(O|\mu)}$$

while the probability of being in state $i$ at time $t$, given the observed sequence and model is the following and can be computed using the forward and backward variables:

$$\xi_t(i,j) = \frac{a_i(t)[x_i,x_j]b[x_j,o_{t+1}]\beta_j(t+1)}{Pr(O|\mu)}$$

The expected number of transitions from state $i$ to state $j$ compared to the expected total number of transitions away from state $i$:

$$\bar{\alpha}[x_i,x_j] = \frac{\sum_{t=1}^{T-1}\xi_t(i,j)}{\sum_{t=1} T-1\gamma_j(t)}$$

An alternative simple fitting algorithm for HMMs is Viterbi training. In this method, each observation is tagged with the most likely state to generate it using the Viterbi algorithm [20] [21] where the goal is to find the most probable sequence of states – also known as max-sum algorithm. The distributions (emissions) of each states are then updated using MLE estimates on the observations which were generated from them, and the transition matrix is updated by looking at pairs of adjacent state taggings. Concluding, the Viterbi algorithm generates the final most probable path amongst all the possible ones.

However, this method is not optimal because it only calculates the single most likely path. Baum-Welch on the other hand uses the probabilities of all paths and does not use hard assignments. For our HMMs, we utilize either Baum-Welch or we estimate the transition probabilities by simply counting transition occurrences between states for exclusively supervised scenarios – also referred to as 'simple MLE estimation' in literature.

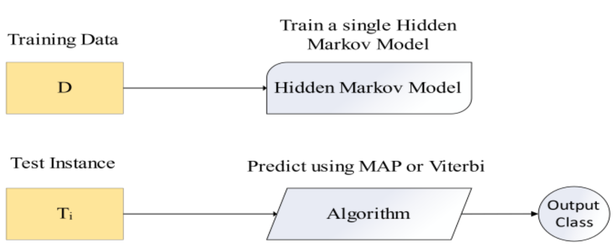


Fig. 2. First examined approach for training and testing HMMs

## IV. Experimental Study

An experimental study was performed in order to examine the performance of the Hidden Markov Models. Initially, public datasets that are widely used for sentiment analysis were selected in the context of the study. Then, in the second part of the study, the performance in specifying user sentiment is evaluated.

### A. Data Used

A collection of benchmark datasets which consist of sequences of sentences forming a document were selected and used in the experimental procedures. It is necessary for the datasets to be annotated at sentence or phrase level and as a result two datasets were selected, the Fine-grained Sentiment Dataset and the Sentiment Polarity Annotations Dataset.

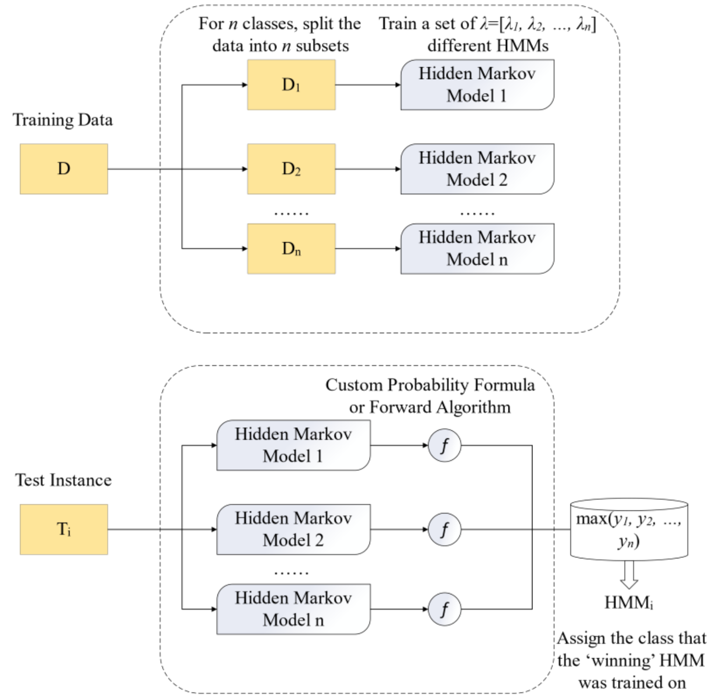


Fig. 3. Second examined approach for training and testing HMMs

The Fine-grained Sentiment Dataset [18] is large corpus of consumer reviews from a range of domains, each review annotated with document sentiment automatically extracted from its star rating, and a small subset of reviews manually annotated at the sentence level with a raw inter-annotator agreement of 86%. There are 5 polarity classes: POS, NEG, NEU, MIX and NR. It was created in 2011 with the reasoning behind its construction being that the authors required a dataset that would be annotated at both the sentence and document levels.

TABLE I. Fine-graned Sentiment Dataset

| | Positive | Negative | Neutral | Total |
|---|---|---|---|---|
| Documents | 97 | 99 | 98 | 294 |
| Sentences | 923 | 1320 | 1593 | 3836 |

The Sentiment Polarity Annotations Dataset (SPOT) dataset [1] was created in 2017 for the task of fine-grained sentiment analysis from the perspective of multiple instance learning. It contains 197 reviews originating from the Yelp 2013 Challenge [22] as well as the Diao et al. corpus [5], annotated with segment-level polarity labels (positive/neutral/negative). Annotations have been gathered on 2 levels of granularity: (1) sentences, (2) sub-sentence clauses produced by a state-of-the-art RST parser named EDUs.

TABLE II. Sentiment Polarity Annotations Dataset

| | From Yelp | | From IMDb | | Total |
|---|---|---|---|---|---|
| | Sentences | EDUs | Sentences | EDUs | |
| Segments | 1065 | 2110 | 1029 | 2398 | 6602 |
| Documents | 100 | | 97 | | 197 |



Authorized licensed use limited to: The Claremont Colleges Library. Downloaded on March 07,2023 at 21:12:56 UTC from IEEE Xplore. Restrictions apply.

The two datasets were used to train the models using different feature sets and evaluate their performance. In the following subsection the characteristics of the evaluation study is presented as well as the experimental setup and the results collected.

### B. Results

A wide range of experiments regarding Machine Learning algorithms as well as a variety of Hidden Markov Models, using k-fold cross-validation was performed. We utilized the same number of sentence label features as there are golden truth labels, which enables the possibility of evaluating the traditional single Hidden Markov Model approach, denoted as Approach A. The algorithms used to train the Hidden Markov Model are the Baum-Welch (referred to as "BW") and the Labeled, as explained in detail in Section III. In particular, for the Baum-Welch algorithm, we limited the number of iterations to one since it is being compared to the simple labeled algorithm.

In the context of the experimental study, a variety of classification algorithms were examined. More specificaly, Naïve Bayes, SVMs and Decision Trees were trained and their performance was compared to the Hidden Markov Models. The features sets that were used in the experimental study were the following: i) Sequence of sentence labels (seq-labels). In the case of baseline Machine Learning algorithms, the sequential information cannot be properly utilized unlike Hidden Markov Models, ii) the combination of sequence labels with the Bag-of-Words model (BoW). The performance of the classifiers on the two datasets is illustrated in the following tables.

TABLE III. PERFORMANCE ON THE FINEGRAINED DATASET

| Method | Features | F1-score | Accuracy |
|---|---|---|---|
| Naive Bayes | seq-labels | 68.609 | 74.333 |
| SVM Linear Kernel | seq-labels | 74.857 | 80.759 |
| Decision Tree | seq-labels | 70.994 | 77.234 |
| Naive Bayes | BoW + seq-labels | 71.555 | 76.966 |
| SVM Linear Kernel | BoW + seq-labels | 75.785 | 80.483 |
| Decision Tree | BoW + seq-labels | 70.442 | 73.133 |
| General Mixture Model BW (Approach A) | sentence labels | 70.121 | 75.31 |
| State-emission HMM BW (Approach A) | sentence labels | 70.841 | 76.126 |
| State-emission HMM BW (Approach B) | sentence labels | **77.435** | **82.759** |
| State-emission HMM Labeled (Approach B) | sentence labels | **77.435** | **82.759** |
| State-emission HMM $2^{nd}$-Order (Approach B) | sentence labels | 76.291 | 79.862 |
| HMM 2-gram (Approach B) | sentence labels | 75.997 | 80.759 |

TABLE IV. PERFROMANCE ON THE SPOT DATASET

| Method | Features | F1-score | Accuracy |
|---|---|---|---|
| Naive Bayes | seq-labels | 69.282 | 78.333 |
| SVM Linear Kernel | seq-labels | 76.175 | 81.034 |
| Decision Tree | seq-labels | 69.303 | 76.667 |
| Naive Bayes | BoW + seq-labels | 71.349 | 78.333 |
| SVM Linear Kernel | BoW + seq-labels | 76.652 | 82.759 |
| Decision Tree | BoW + seq-labels | 71.402 | 81.356 |
| General Mixture Model BW (Approach A) | sentence labels | 67.995 | 73.193 |
| State-emission HMM BW (Approach A) | sentence labels | 69.034 | 73.204 |
| State-emission HMM BW (Approach B) | sentence labels | 78.230 | 82.456 |
| State-emission HMM Labeled (Approach B) | sentence labels | 78.230 | 82.456 |
| State-emission HMM $2^{nd}$-Order (Approach B) | sentence labels | **79.599** | **83.051** |
| HMM 2-gram (Approach B) | sentence labels | 78.432 | 82.531 |

The evaluation results are quite impressive and display interesting findings regarding the performance of all the classifiers. An important finding concerns the superiority of the performance of HMMs which significantly outperform the other classifiers by utilizing the sequential information of the sentence labels. As expected, since we have a supervised task where labels for states are available, Baum-Welch and Viterbi do not provide a performance gain compared to the simple Labeled algorithm. Another finding of the study is that the exclusive to supervised tasks Approach B outperforms Approach A since it also utilizes more of the available labels.

Regarding the baseline machine learning classifiers, when they were trained solely with the Bag-of-Words (BoW) features of the textual data, their performance was quite low. When they were trained based on sentence labels, they achieved better a performance and the results indicate that combining both BoW and sentence features assists in achieving better performance. Indeed, the best performance of the machine learning classifiers was reported by an SVM when trained on both features with an accuracy of 80.483% and an f1-score of 75.785 on Finegrained and an accuracy of 81.759% and an f1-score of 76.452% on the SPOT dataset. It is worth highlighting the superiority of the Hidden Markov Models where the state-emission HMM reported the best performance in both datasets with an accuracy of 82.759% and an F1 score of 77.435% on Finegrained and an accuracy of 83.051% and F1-score of 79.599 on the SPOT dataset respectively.

Overall, the results indicate that the HMMs achieve robust performance and steadily outperform the baseline machine learning classifiers. The performance of the HMMs is up to 3.6% higher compared to the performance of the SVM which was the best performing machine learning classifier.



Authorized licensed use limited to: The Claremont Colleges Library. Downloaded on March 07,2023 at 21:12:56 UTC from IEEE Xplore. Restrictions apply.

## V. CONCLUSIONS

In this work we proposed a variety of Hidden Markov Models for a sentiment analysis task, which can also be applied to any text classification tasks. Product reviews and user-generated content in social media contain very useful information about public opinion and in order to perform analysis and extract knowledge, automated processing methods are required, which can be highly challenging due to the complexity of natural language. Recognizing emotions, sentiments and attitudes in textual data can greatly assist in understanding peoples' opinions and is highly desired for in various procedures and applications. Given the special characteristics of the task of sentiment analysis and textual emotion recognition which is the sequential nature of data, utilizing HMMs seems to be a suitable and potent approach. Indeed, the utilization of HMMs can increase the recognition performance since they take into account the sequential nature of the textual data, a very important piece of information that traditional machine learning methods neglect and fail to take into account. To the best of our knowledge, such a wide variety of HMMs has not been thoroughly studied for text classification tasks.

We perform a variety of experiments on HMMs and expand on their usage for text classification and its challenges. Evaluation results show that the proposed models achieve high accuracy on quite small datasets and outperform machine learning approaches that cannot utilize the sequential nature of the textual data. The results are quite interesting since the high performance indicates that the Hidden Markov Model is scalable as well as accurate in specifying users' opinions and attitudes. Furthermore, our proposed architecture of training a HMM for each class label of the dataset outperforms the traditional single HMM approach, showcasing that it is effectively applicable to classification tasks and is in fact the superior approach for supervised scenarios. Such a comparison between these two architectures was ignored by previous work on the field. As a result, HMMs are a potent tool for natural language processing as well as Sentiment Analysis tasks, which can utilize the sequential nature of textual data.

There are various directions that future work could explore. An interesting direction concerns the utilization of weighted voting ensembles of Hidden Markov Models and the examination of their performance under different training parameters and ensemble schemas. This constitutes a main direction of our future work.

## ACKNOWLEDGMENT

This work was partially supported by the project "Strengthening the Research Activities of the Directorate of the Greek School Network and Network Technologies", funded by the Computer Technology Institute and Press "Diophantus" with project code 0822/001.

## REFERENCES

[1] S. Angelidis, and M. Lapata, "Multiple Instance Learning Networks for Fine-Grained Sentiment Analysis", Transactions of the Association of Computational Linguistics, 6, pp. 17-31, 2018

[2] L. Augustyniak, T. Kajdanowicz, P., Szymanski, W., Tuliglowicz, P., Kazienko, R. Alhajj, and B., Szymanski,. "Simpler is better? Lexicon-based ensemble sentiment classification beats supervised methods. In Advances in Social Networks Analysis and Mining (ASONAM), IEEE/ACM International Conference on, pp. 924-929. 2014, IEEE.

[3] C. M. Bishop, "Pattern recognition and machine learning", 2006, Springer.

[4] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis", IEEE Intelligent Systems, vol 28(2), pp. 15-21, 2013.

[5] Q. Diao, M. Qiu, C.Y. Wu, A.J. Smola, J. Jiang, and C. Wang, "Jointly modeling aspects, ratings and sentiments for movie recommendation", In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining , pp. 193-202, 2014, ACM.

[6] F. Hemmatian, and M. K. Sohrabi, "A survey on classification techniques for opinion mining and sentiment analysis", Artificial Intelligence Review, pp. 1-51. 2017.

[7] D. M. Hussein, "A survey on sentiment analysis challenges", Journal of King Saud University-Engineering Sciences, vol *30*(4), pp. 330-338, 2018.

[8] E. L. Iglesias, L. Borrajo, and R. Romero, "A HMM text classification model with learning capacity". *ADCAIJ:* Advances in Distributed Computing and Artificial Intelligence Journal, 3(3), 21, 2014

[9] M. Kang, J.Ahn, and K. Lee "Opinion mining using ensemble text hidden Markov models for text classification", *Expert Systems with Applications,* vol 94, pp. 218-227, 2018.

[10] L. Liu, D. Luo, M. Liu, J. Zhong, Y. Wei, and L. Sun, "A self-adaptive hidden markov model for emotion classification in chinese microblogs", Mathematical problems in engineering, 2015.

[11] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey", Ain Shams Engineering Journal, vol 5(4), pp. 1093-1113, 2014.

[12] I. Perikos, and I. Hatzilygeroudis, "A framework for analyzing big social data and modelling emotions in social media", In 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService), pp. 80-84. March 2018, IEEE.

[13] I. Perikos, and I. Hatzilygeroudis, "Recognizing emotions in text using ensemble of classifiers", Engineering Applications of Artificial Intelligence, *vol 51*, pp. 191-201, 2016.

[14] I. Perikos, and I. Hatzilygeroudis, "Recognizing emotion presence in natural language sentences", In International conference on engineering applications of neural networks, pp. 30-39, September 2013, Springer, Berlin, Heidelberg.

[15] C. Quan, and F. Ren, "Weighted high-order hidden Markov models for compound emotions recognition in text", Information Sciences, vol 329, pp. 581-596, 2016.

[16] S. Shaheen, W. El-Hajj, H. Hajj, and S. Elbassuoni, "Emotion Recognition from Text Based on Automatically Generated Rules", In Data Mining Workshop (ICDMW), 2014 IEEE International Conference on, pp. 383-392. 2014, IEEE.

[17] S. Soni, and A. Sharaff, "Sentiment Analysis of Customer Reviews based on Hidden Markov Model", In *Proceedings of the 2015 International Conference on Advanced Research in Computer Science Engineering & Technology (ICARCSET 2015)* pp. 12, 2015, ACM.

[18] O. Täckström, and R. McDonald, "Discovering fine-grained sentiment with latent variable structured prediction models", In European Conference on Information Retrieval pp. 368-374, April 2011, Springer, Berlin, Heidelberg.

[19] G. Vinodhini, and R. M. Chandrasekaran, "Sentiment analysis and opinion mining: a survey", International Journal, 2(6), 2012.

[20] A. Viterbi, "A personal history of the Viterbi algorithm", IEEE Signal Processing Magazine, vol. 23(4), pp. 120-142, 2006.

[21] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm", IEEE transactions on Information Theory, vol 13(2), pp. 260-269. 1967.

[22] Yelp Dataset Challenge - https://www.yelp.com/dataset/challenge

[23] X. Zhao, and Y. Ohsawa, "Sentiment Analysis on the Online Reviews Based on Hidden Markov Model", Journal of Advances in Information Technology, vol 9(2), pp. 33-38, 2018.



Authorized licensed use limited to: The Claremont Colleges Library. Downloaded on March 07,2023 at 21:12:56 UTC from IEEE Xplore. Restrictions apply.