Sentiment analysis using novel and interpretable architectures of Hidden Markov Models

Wenhao Xue, Zihan Liu, Junjie Wang

1 Introduction

With the rapid development of the Internet and the widespread use of social media, people are increasingly expressing their opinions and emotions online. Understanding and analyzing user sentiments on social media is of great significance for grasping the public's emotional tendencies towards events, products, services, etc. Sentiment analysis aims to automatically identify, extract, and analyze people's sentiments, opinions, and attitudes from textual data. It has broad application prospects in areas such as public opinion analysis, product reviews, and political elections.

Existing sentiment analysis methods mainly include lexicon-based methods and machine learning-based methods. Lexicon-based methods rely on sentiment dictionaries, but constructing sentiment dictionaries requires a large amount of human and material resources and lacks consideration of context. Although machine learning-based methods have achieved good results, they cannot effectively utilize the sequential information of text and lack interpretability.

To address the limitations of existing methods, this paper explores the use of Hidden Markov Models (HMMs) for sentiment analysis. HMMs are naturally suitable for processing sequential data and can fully utilize the contextual relationships between words in text. This paper proposes a novel interpretable HMM-based sentiment analysis method and conducts systematic research on model architectures, training approaches, higher-order HMMs, and model ensembles.

Lexicon-based approaches rely heavily on sentiment lexicons, which are utilized in order to represent predetermined and precompiled, negative and positive words.

Lexicon	Positive Words	Negative Words
Simplest (SM)	good	bad
Simple List (SL)	good, awesome, great, fantastic, wonderful	bad, terrible, worst, sucks, awful, dumb
Simple List Plus (SL+)	good, awesome, great, fantastic, wonderful, best, love, excellent	bad, terrible, worst, sucks, awful, dumb, waist, boring, worse
Past and Future (PF)	will, has, must, is	was, would, had, were
Past and Future Plus (PF+)	will, has, must, is, good, awesome, great, fantastic, wonderful, best, love, excellent	was, would, had, were, bad, terrible, worst, sucks, awful, dumb, waist, boring, worse
Bing Liu	2006 words	4783 words
AFINN-96	516 words	965 words
AFINN-111	878 words	1599 words
enchantedlearning.com	266 words	225 words
MPAA	2721 words	4915 words
NRC Emotion	2312 words	3324 words

Figure 1: Sentiment Lexicons

The main contributions of this paper include:

1.proposing a new interpretable HMM-based sentiment analysis method that can reveal the internal decision-making process of the model;

2.exploring different HMM model architectures and training methods, and conducting detailed experimental comparisons;

3.investigating the application of higher-order HMMs in sentiment analysis;

4.proposing to integrate multiple HMMs for ensemble learning to further improve performance.

Experimental results show that the proposed HMM-based sentiment analysis method can achieve better performance than traditional machine learning methods and has good interpretability.

2 Methodology

2.1 Hidden Markov Model

A Hidden Markov Model (HMM) is essentially a statistical model where the system is considered a Markov process with unobservable, or hidden, states. It is a probabilistic framework that assigns probabilities to different sequences of labels, essentially functioning as a general mixture model that includes transition matrices. These hidden states form a Markov chain characterized by specific transition probabilities and adhere to the Markov property, implying that the state's dependency is only on its immediate predecessor, reflecting a property of memorylessness. Therefore, in predicting the next item in a sequence, the model relies solely on the most recent observation, excluding all others. The hidden states in this model are discrete, while the observations themselves may be either continuous or discrete.

An HMM is a tuple $\theta = (X, O, \pi, A, B)$ where:

$X = \{x_1, x_2, ..., x_n\}$ is a set of elements that are called states.

$O = \{o_1, o_2, ..., o_m\}$ is a set of observations.

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

A is the transition probabilities matrix. The size is $n \times n$. and $a_{ij} = P(x_i|x_j)$ represents the transition probability from the hidden state i to the hidden state j. An alternative notation for a_{ij} is $a[x_i, x_j]$.

B is the observation probabilities sequence. Each $b_i(o_i) = P(o_i|x_i)$ represents the probability of an observation x_i being generated from a state x_i. An alternative notation for $b_i(o_i)$ is $b[x_i, o_i]$.

HMMs feature a variety of advantages. A main strength that the HMMs possess is that they have the ability to model sequences of varying lengths. Furthermore, HMMs showcase a certain degree of invariance when it comes to time axis warping.

2.2 High-order Hidden Markov Models

The traditional HMM and the Markov chain it is based on, depend only on the value of the immediately preceding observation and are independent of all earlier observations. This is very restrictive.

One way to allow more than one of the previous observations to have an influence on the model is to move to higher-order Markov chains. Any higher-order HMM, or Markov chain, can be transformed into an equivalent general first-order HMM/process and traditional training algorithms can be applied for training.

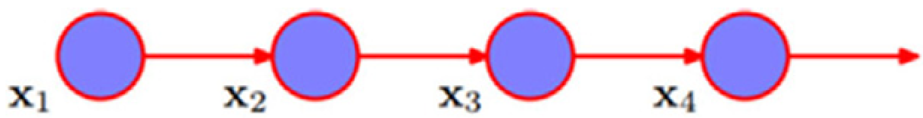

Figure 2: A first-order Markov Chain.

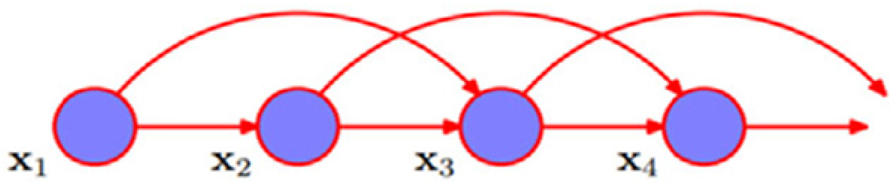

Figure 3: A second-order Markov Chain.

For a traditional Markov chain, referred to as first-order, the joint distribution for a sequence of N observations, $x_1, x_2, ..., x_N$ is given by:

$$p(x_1, \ldots, x_N) = p(x_1) \prod_{n=2}^{N} p(x_n|x_{n-1})$$

In Figure 2, the first-order Markov chain is illustrated.

For a second-order Markov chain, an observation depends on the values of the two previous observations.

$$p(x_1, \ldots, x_N) = p(x_1)p(x_2|x_1) \prod_{n=3}^{N} p(x_n|x_{n-1}, x_{n-2})$$

For example, the observation of x_3 depends on the values of x_2 and x_1 as illustrated in Figure 3.

In the same context, we can create higher-order HMMs instead of first-order HMMs. An n-order HMM takes into account longer past state sequences and is represented by a tuple θ similar to traditional HMMs except for the transition probabilities $O^1, O^2, ..., O^n$, which is a set of transition matrices, and the initial probabilities.

2.3 Training phase

In the HMM implementations that we introduce, we make use of either Baum Welch, also called Maximum Likelihood Estimation method. This algorithm is based on the expectation maximization theorem, which given a selection of observed feature vectors, attempts to find the MLE of the parameters of a HMM.

The stages of the main procedures is:

1.Initiate with initial probability estimates referring to a model θ.

2. Calculate the expectations of how often each emission and transition is used, corresponding to the parameters of model θ.

3. Implement proper changes to the model by maximizing paths.

4. Attempt to estimate the probabilities iteratively until convergence is reached.

2.4 Prediction phase

For the evaluation (prediction) phase, the two main options are the Viterbi path and the maximum a-posteriori (MAP) method that is also known as the forward–backward algorithm. The Viterbi path is the most likely sequence of states that generated the sequence, given the full model. The MAP calculates the most likely state per observation in the sequence given the entire remaining alignment.

We used those two algorithms in our first approach, called Approach A (see Figure 4), where a single HMM is used for training.

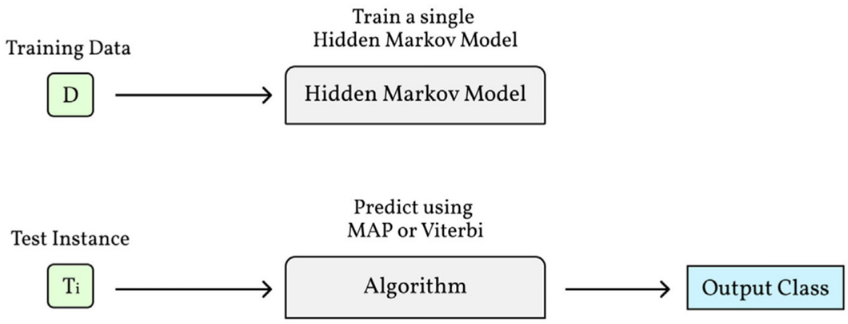

Figure 4: Approach A for training and testing HMMs.

However, the prediction phase is significantly different when utilizing an alternative approach used in classification tasks in which an HMM model is trained for each of the class labels. We name this architecture Approach B (see Figure 5). When a new instance arrives, we calculate the probability of the instance being generated by each of the models using a custom formula. The instance is labeled with the class associated with the maximum probability, i.e. the model that was most likely to have generated it.

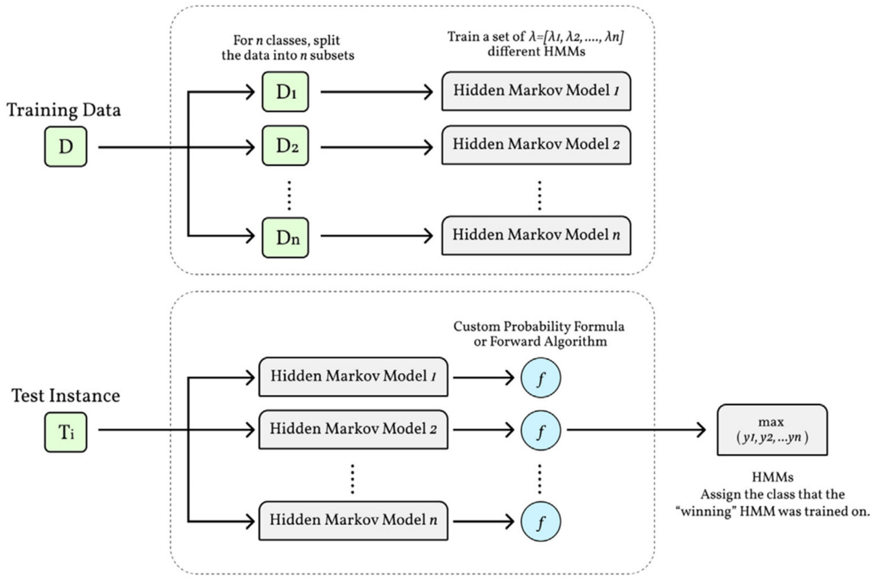

Figure 5: Approach B for training and testing HMMs.

2.5 Implementation

Regarding the implementations of the HMMs, we first present two of the main challenges that need to be faced.

The first challenge that needs to be faced concerns the high feature space dimensionality that leads to issues on the transition probabilities matrix. We present two approaches in the experiments to solve the problem: "Clustering Solver" and "Artificial Solver".

The second concerns encountering out-of-vocabulary (OOV) new observations that do not exist in the probability matrix. The solution is to utilize a smoothing factor also referred to as emission pseudo-count as the probability estimate of out-of vocabulary observations.

2.6 Ensemble learning

In general, ensemble learning aims to improve the performance of individual methods by combining learning algorithms. Ensembles combine hypotheses with the aim of forming an even better solution.

The output class of a given instance is determined by the weighted vote of the log probability of the multiple models combined in the ensemble

$$\log(p\,(1)), \log(p\,(2)), \ldots, \log(p\,(n)), where \sum_{i=0}^{n} p\,(i) = 1$$

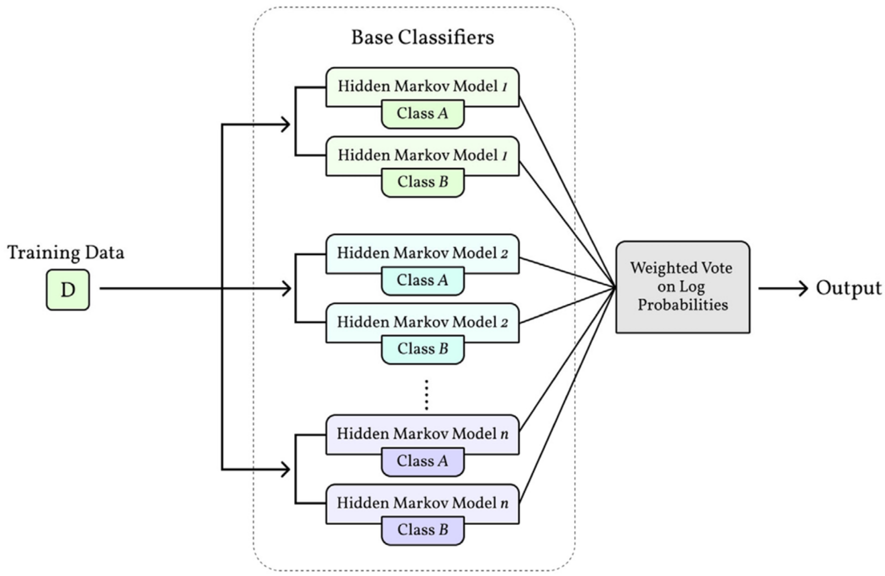

Figure 6: The overview of the proposed ranked weighted vote ensemble scheme.

3 Result

3.1 Design

A concrete experiment was designed and conducted with the aim to assess the performance of the HMM and study their interpretability. Publicly available datasets were utilized and different types of textual data were used to assess the performance of the methods and also provide a deeper insight into their performance on heterogeneous data from different sources.

We used the following data: Fine-Grained Sentiment Dataset, the Sentiment Polarity Annotations Dataset(SPOT), the Movie Review Polarity(MR), the Movie Review Subjectivity(SUBJ) and the IMDb Large Movie Review Dataset.

	Documents				Sentences			
	Positive	Negative	Neutral	Total	Positive	Negative	Neutral	Total
Books	19	20	20	59	160	195	384	739
DVDs	19	20	20	59	164	264	371	799
Electronics	19	19	19	57	161	240	227	628
Music	20	20	19	59	183	179‘	276	638
Video games	20	20	20	60	255	442	335	1032
Total	**97**	**99**	**98**	**294**	**923**	**1320**	**1593**	**3836**

Figure 7: Fine-grained sentiment dataset.

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

Figure 8: Sentiment polarity annotations dataset.

	MR	SUBJ	IMDB
Instances	10662	10000	50000

Figure 9: Distribution of the instances of the MR, SUBJ, and IMDB datasets.

3.2 Results on low dimensionality datasets

The objective of the conducted experiments is to assess the performance of models under various feature sets, training parameters and architectures in low feature count scenarios. The algorithms used to train the HMMs are Baum Welch algorithm (referred to as "BW") and the Labeled algorithm.

For the training procedure, the feature sets used in the experiments were:

(i) The words themselves, also known as the Bag-of-Words (BoW) model.

(ii) The sequence of labels of the sentences (noted as seqlabels).

(iii) Both of the above feature sets.

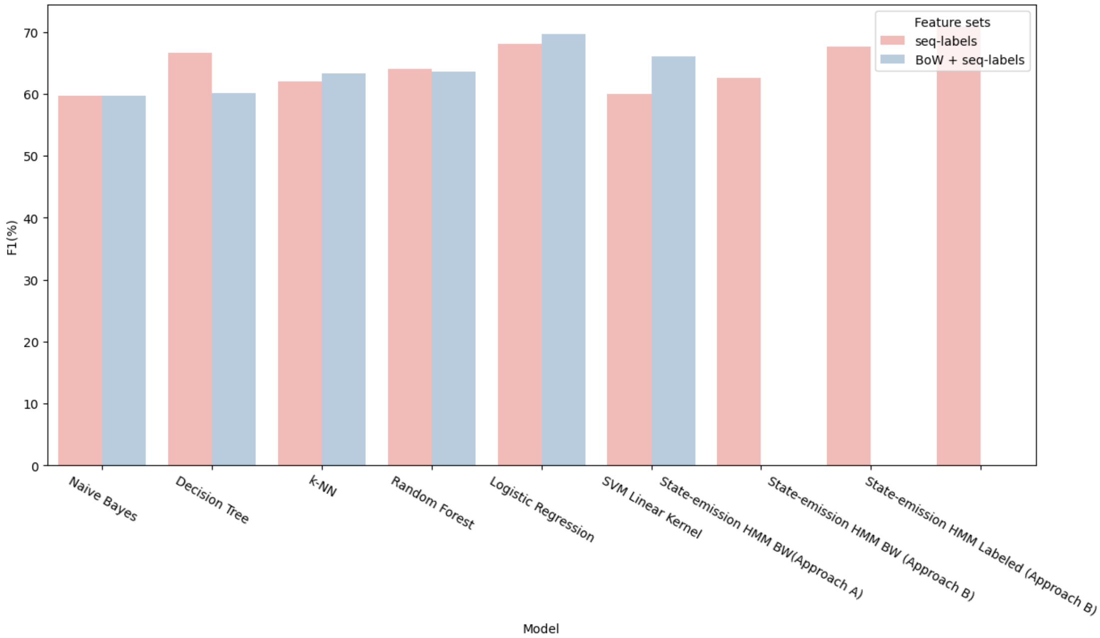

Figure 10: SPOT SENTENCE(ML&HMM)

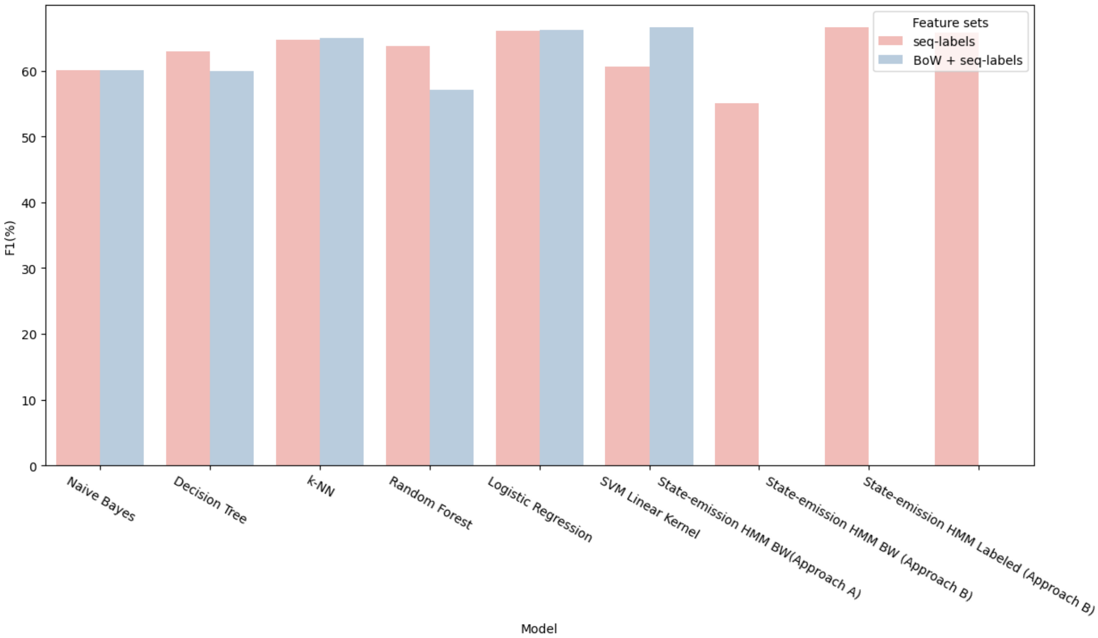

Figure 11: SPOT EDUs(ML&HMM)

Table 1: spot sentences and EDUs

Model	Feature sets	Sentences	EDUs
Naive Bayes	seq-labels	59.634	60.02
Decision Tree	seq-labels	66.58	62.9
k-NN	seq-labels	62.019	64.63
Random Forest	seq-labels	64.023	63.741
Logistic Regression	seq-labels	68.013	66.036
SVM Linear Kernel	seq-labels	59.95	60.581
Naive Bayes	BoW + seq-labels	59.634	60.015
Decision Tree	BoW + seq-labels	60.068	60
k-NN	BoW + seq-labels	63.369	64.994
Random Forest	BoW + seq-labels	63.537	57.124
Logistic Regression	BoW + seq-labels	69.688	66.124
SVM Linear Kernel	BoW + seq-labels	66.019	66.537
State-emission HMM BW(Approach A)	seq-labels	62.514	55.114
State-emission HMM BW (Approach B)	seq-labels	67.7	66.596
State-emission HMM Labeled (Approach B)	seq-labels	70.813	65.728
State-emission HMM 2nd-Order (Approach B)	seq-labels	67.829	32.387
State-emission HMM 3rd-Order (Approach B)	seq-labels	66.724	32.387
State-emission HMM 4th-Order (Approach B)	seq-labels	60.247	-
Ensemble of 3 Best nth-Order HMMs (Approach B, Sum)	seq-labels	69.816	71.217
Ensemble of 3 Best nth-Order HMMs (Approach B, Weighted Sum)	seq-labels	70.342	69.852
Ensemble of 3 Best nth-Order HMMs (Approach B, Product)	seq-labels	70.402	69.34
Ensemble of 3 Best nth-Order HMMs (Approach B, Borda count)	seq-labels	69.769	69.793

The results highlight the better performance of the HMMs. The main reason :they use appropriately the sentence sequence labels, a piece of information that machine learning algorithms cannot properly take into account.

3.3 Results on high dimensionality datasets

Datasets that consist of many instances something that results in high dimensionality in terms of features for the HMM based methods. The best performance of the HMMs models is achieved again by the Artificial Solver . The Artificial Solver approach has the potential to take any base machine learning classifier and increase its performance by utilizing the sequential information of the text. It is worth to notice that utilizing higher-order HMMs leads to higher performance until the third-order on large datasets. The result show that the larger the datasets are, the higher the performance of high-order HMMs is.

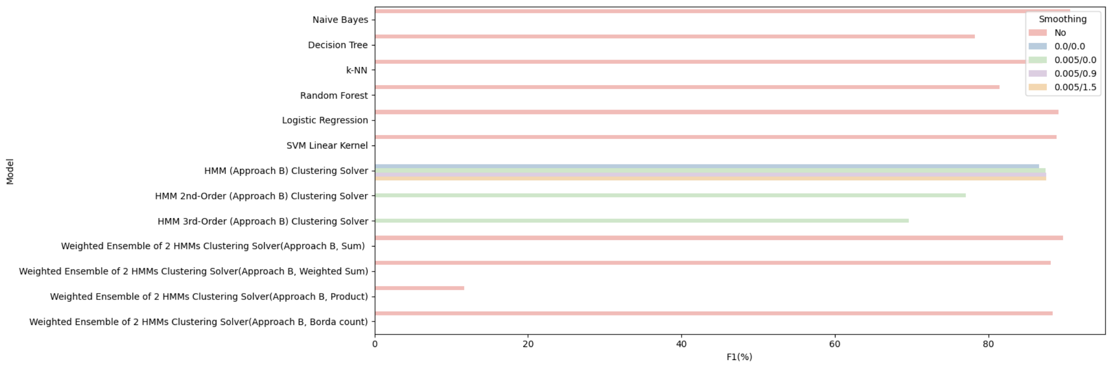

Figure 12: SUBJ(ML&HMMs)

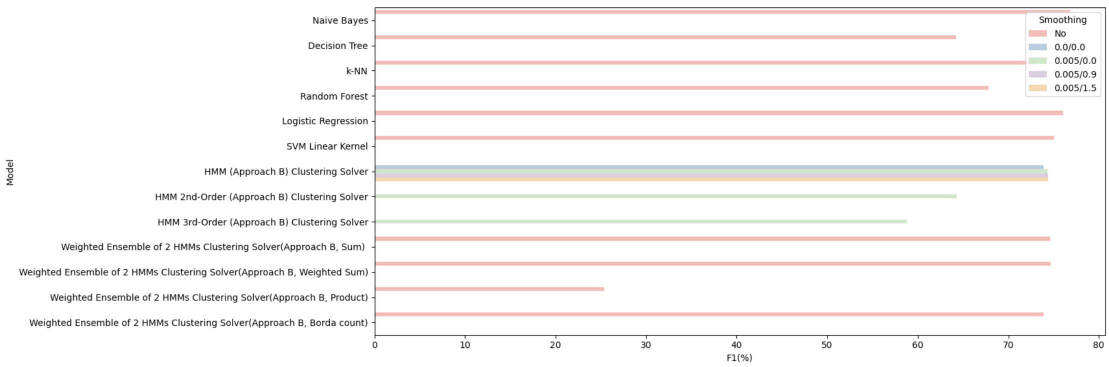

Figure 13: MR(ML&HMMs)

Table 2: SUBJ and MR

Model	Smoothing	SUBJ	MR
Naive Bayes	-	90.659	76.842
Decision Tree	-	78.215	64.243
k-NN	-	86.998	72.9
Random Forest	-	81.452	67.807
Logistic Regression	-	89.139	76.075
SVM Linear Kernel	-	88.859	75.063
HMM (Approach B) Clustering Solver	0.0/0.0	86.604	73.873
HMM (Approach B) Clustering Solver	0.005/0.0	87.434	74.349
HMM (Approach B) Clustering Solver	0.005/0.9	87.506	74.372
HMM (Approach B) Clustering Solver	0.005/1.5	87.532	74.405
HMM 2nd-Order (Approach B) Clustering Solver	0.005/0.0	77.015	64.32
HMM 3rd-Order (Approach B) Clustering Solver	0.005/0.0	69.658	58.769
Weighted Ensemble of 2 HMMs Clustering Solver(Approach B, Sum)	-	89.69	74.583
Weighted Ensemble of 3 HMMs Clustering Solver(Approach B, Weighted Sum)	-	88.087	74.678
Weighted Ensemble of 4 HMMs Clustering Solver(Approach B, Product)	-	11.713	25.367
Weighted Ensemble of 5 HMMs Clustering Solver(Approach B, Borda count)	-	88.388	73.882
State-emission HMM 3rd-Order (Approach B)	seq-labels	66.724	32.387
State-emission HMM 4th-Order (Approach B)	seq-labels	60.247	-
Ensemble of 3 Best nth-Order HMMs (Approach B, Sum)	seq-labels	69.816	71.217
Ensemble of 3 Best nth-Order HMMs (Approach B, Weighted Sum)	seq-labels	70.342	69.852
Ensemble of 3 Best nth-Order HMMs (Approach B, Product)	seq-labels	70.402	69.34
Ensemble of 3 Best nth-Order HMMs (Approach B, Borda count)	seq-labels	69.769	69.793

In general, the bigger a dataset is and the longer the sentences are, the more potent the high order HMMs can be.

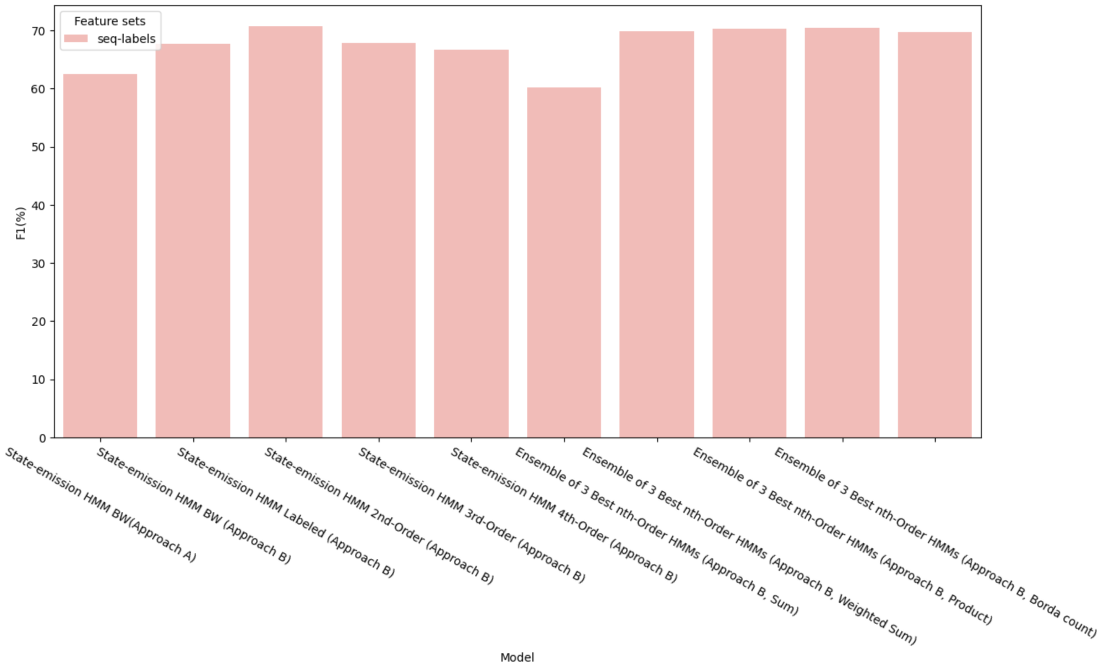

Figure 14: SPOT Sentence(HMMs)

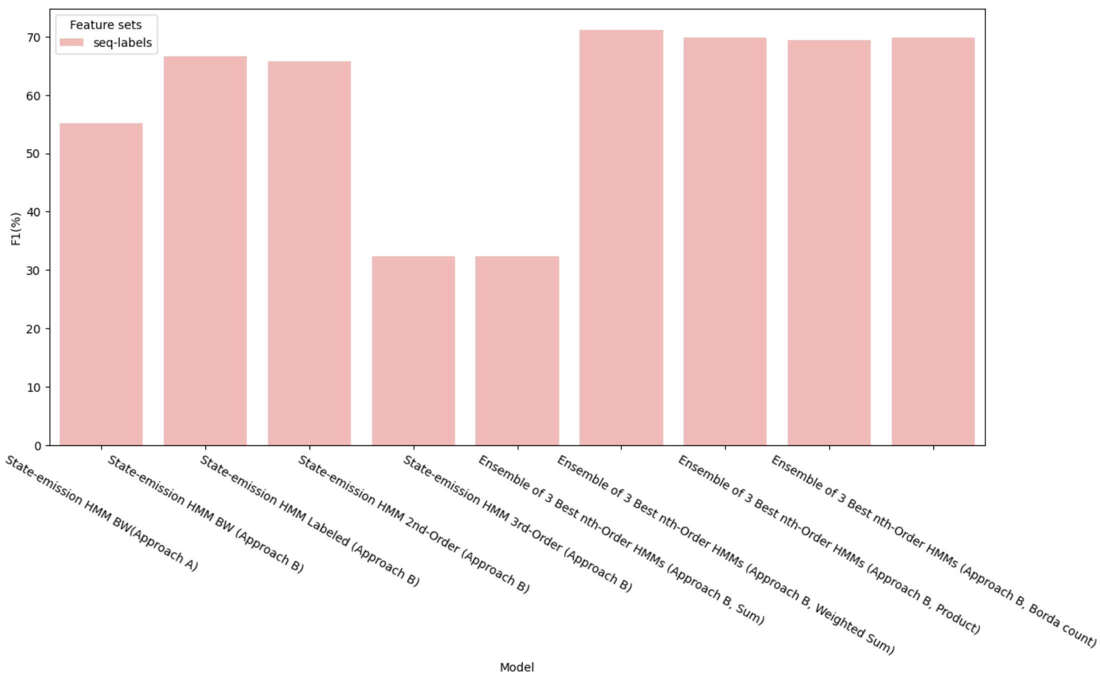

Figure 15: SPOT EDUs(HMMs)

4 Conclusions and Discussion

This paper introduced a novel and interpretable sentiment analysis method using Hidden Markov Models (HMMs). Through systematic research and comparisons of different HMM architectures and training approaches, we demonstrated the efficacy of HMMs in processing sequential text data. Our experimental results indicate that our HMM-based sentiment analysis method outperforms traditional machine learning approaches, offering enhanced performance and interpretability.

Our study confirms the advantages of higher-order HMMs in capturing long-distance dependencies and shows that performance can be further improved by integrating multiple HMMs into an ensemble. Additionally, our approach provides insights into the internal decision-making process of the model, which is crucial for explaining predictions and gaining user trust.

Despite the promising outcomes, there are several limitations and challenges associated with our method. First, the computational complexity of higher-order HMMs may limit their application on large-scale datasets. Second, while our model offers interpretability, further enhancing the transparency and understandability for end users remains a challenge.

Moreover, the accuracy of sentiment analysis largely depends on the quality and representativeness of the training data. In the future, we plan to explore more data preprocessing and augmentation techniques to improve the robustness of our model against imbalanced or biased data.

5 Contributions

Wenhao Xue: The writing of the paper.
Zihan Liu: Presentation.
Junjie Wang: Writing code and icon making.

References

[1] Isidoros Perikos, Spyridon Kardakis, Ioannis Hatzilygeroudis, Sentiment analysis using novel and interpretable architectures of Hidden Markov Models, Knowledge-Based Systems, Volume 229, 2021, 107332, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2021.107332.

[2] L.E. Baum, T. Petrie, G. Soules, N. Weiss, A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, Ann. Math. Stat. 41 (1) (1970) 164–171.

[3] G.D. Forney, The viterbi algorithm, Proc. IEEE 61 (3) (1973) 268–278.

[4] H.W. Sorenson, Parameter estimation: principles and problems, 1980.

[5] O. Täckström, R. McDonald, Discovering fine-grained sentiment with latent variable structured prediction models, in: Proceedings of the European Conference on Information Retrieval, Springer, 2011, pp. 368–374.

[6] S. Angelidis, M. Lapata, Multiple instance learning networks for fine grained sentiment analysis, Trans. Assoc. Comput. Ling. 6 (2018) 17–31.

[7] B. Pang, L. Lee, Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales, in: Proceedings of the 43rd annual meeting on association for computational linguistics, 2005, pp. 115-124.

[8] B. Pang, L. Lee, A sentimental education: Sentiment analysis using subjec tivity summarization based on minimum cuts, in: Proceedings of the 42nd annual meeting on Association for Computational Linguistics, 2004, p. 271.

[9] A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, C. Potts, Learning word vectors for sentiment analysis, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Lan guage Technologies, 1, Association for Computational Linguistics, 2011, pp. 142–150.