Computers, Materials & Continua
DOI:10.32604/cmc.2021.012304
Article

Tech Science Press

Improving Language Translation Using the Hidden Markov Model

Yunpeng Chang[1], Xiaoliang Wang[1,*], Meihua Xue[1], Yuzhen Liu[1] and Frank Jiang[2]

[1]School of computer science and engineering, Hunan University of Science and Technology, Xiang, 411201, China
[2]School of Info Technology, Deakin University, Geelong, 3220, Australia
[*]Corresponding Author: Xiaoliang Wang. Email: fengwxl@hnust.edu.dn

Received: 30 August 2020; Accepted: 30 December 2020

Abstract: Translation software has become an important tool for communication between different languages. People's requirements for translation are higher and higher, mainly reflected in people's desire for barrier free cultural exchange. With a large corpus, the performance of statistical machine translation based on words and phrases is limited due to the small size of modeling units. Previous statistical methods rely primarily on the size of corpus and number of its statistical results to avoid ambiguity in translation, ignoring context. To support the ongoing improvement of translation methods built upon deep learning, we propose a translation algorithm based on the Hidden Markov Model to improve the use of context in the process of translation. During translation, our Hidden Markov Model prediction chain selects a number of phrases with the highest result probability to form a sentence. The collection of all of the generated sentences forms a topic sequence. Using probabilities and article sequences determined from the training set, our method again applies the Hidden Markov Model to form the final translation to improve the context relevance in the process of translation. This algorithm improves the accuracy of translation, avoids the combination of invalid words, and enhances the readability and meaning of the resulting translation.

Keywords: Translation software; hidden Markov model; context translation

1 Introduction

Language translation has undergone a major change in recent years. Traditional statistical machine translators have considered only the linear relationship between words and neglected sentence structure and context. Differences in word order between languages limited the overall translation performance of these methods. However, rapid developments in deep learning have advanced translation toward intelligence. As a specific type of machine learning, deep learning offers great performance and flexibility [1,2]. Deep learning is capable of describing complex functions of high-order abstract concepts, solving artificial intelligence tasks such as target recognition, voice perception, and voice recognition. In terms of language translation, the performance of neural machine translation (NMT) far surpasses traditional statistical machine methods. Researchers are continuously exploring and optimizing context-based translation methods.

The older mathematical models for statistical machine translation are five word-to-word models originally proposed by IBM researchers, termed IBM model 1 to IBM model 5. Google's earlier online translation system was based on statistical machine translation. The system created a translation corpus by searching a large number of bilingual web pages, selected the most common correspondence between words, and generated translation results according to the mathematical model. The Internet now provides an abundant corpus, providing a foundation for the development and improvement of statistics-based machine translation methods. A few years ago, Google began to use a recurrent neural network (RNN) for translation, directly learning mappings from an input sequence (such as a sentence in one language) to an output sequence (the same sentence in another language). Existing NMT approaches are constrained by NMT's one-way decoder. A one-way decoder cannot predict target words according to the context to be generated, but only according to historical information. However, the dependency between words is uncertain, and historical information may not be sufficient to predict the target words using NMT. The quality of translation is greatly influenced by the dependencies between words [3]. However, NMT does treat the whole input sentence as the basic unit of translation.

We present a translation method using the Hidden Markov Model (HMM) combined with context. The input text is processed by a Hidden Markov Model with a phrase-based translation unit. Then, machine learning calculates the sequence of articles with the sentence as the translation unit again to improve the accuracy of translation.

2 Related Work

Machine translation has flourished since its emergence. With the help of a growing corpus, automatic translation has advanced from low-quality results that do not pay attention to grammatical analysis to higher-quality results from analyzing sentence structure and grammar. At present, improving the quality and efficiency of machine translation remains a difficult problem. It is worth exploring better methods of translation that incorporate context.

We offer a new opinion about translation algorithms. In the beginning, methods followed a one-to-one direct translation mode, which established the foundation for the noise channel theory of statistical machine translation as well as the groundwork for future intelligent translation. At that time, the theory took a big step forward in machine translation, but it was just the beginning in terms of language order and structure [4]. The original machine translation technology was put forward in the twentieth century, combining phrase structure grammar with the principles for "generating sentence rules," making rule-based machine translation technology widely popular. Three mainstream rule-based translation methods came to dominate literal translation, transformational translation, and intermediate language translation. Literal translation is the simplest, converting words directly and then rearranging the target results according to rules, but it ignores the overall structure of the sentence. Transformational translation considers not only the correspondence between the two languages in the simple sense, but also the correspondence in the grammatical structure of the sentence. It analyzes the meaning of the sentence according to morphology, grammar, and semantics, which are more advanced than the literal translation. Interlanguage language translation adopts a compromise method that not only considers the multi-level meaning of the sentence, but also ignores it as much as possible in order to create a relatively simple intermediate representation that is then translated into the final language. This approach is less complex while remaining effective. All three methods require the corresponding corpora of the input and output languages, as well as high degree of correspondence in meaning and structure of the input and output languages. Deficiencies here greatly affect the results. We also note that rule-based translation processes are readily explained and more intuitive [5]. The first neural machine translation approach appeared during this time as well, with a proposal from Bahdanau et al. [6].

Japanese translation experts proposed case-based machine translation using a source language instance sentence library. Translation takes place by comparing input sentences to examples in the database and outputting the translations for the most similar ones. The target sentence is then processed further to obtain the final translation. However, the amount of memory available for translation and the system's coverage of a language determine the quality of results. Further, not all users use the same definition of similarity. Therefore, case-based translation requires more effort to be successful [7].

By 2014, the popularity of the Seq2Seq model rose quickly, capturing long-distance information. In this method, the encoder compresses an entire input sentence into a vector with fixed dimensions, with the decoder generating the output sentence according to this vector. The addition of an integrating attention mechanism improves the feature learning ability for long sentences and strengthens the representation ability of source language sequences [8]. Later, the Phrase-Based (PB) model works by dividing sentence x into phrases (word sequences), with each source phrase $x{\sim}ax$ transformed into a target phrase y which are reordered to form the target sentences [9].

Chen et al. [10] proposed that using convolutional neural networks (CNNs) to make the source topic information a potential topic representation on the source statement that the source topic idea of each sentence is learned by the machine according to the word and topic contexts, which is then used to calculate an additional topic context vector to predict the target word. Liu et al. [11] proposed a unified framework for integrating translation memory (TM) into phrase-based statistical machine translation (SMT), enabling the use of global context implicitly and briefly through the local dependency model.

Additionally, in 2014, a graph-based method was proposed by Narouel et al. [12]. This method uses a large body of multilingual vocabulary knowledge (called babelnet) to eliminate the ambiguity in any language, obtains the word meaning clues from each language, and finally connects these clues to obtain the meaning of the target word. After the introduction of the phrase-based statistical machine translation model, Peris et al. [13] proposed corpus-based technology using a neural network. Bengio et al. [14] used a neural network classifier to deal with the sparsity and nonlinearity of features for different neuro-linguistic programming (NLP) tasks. This neural network extracts embedded features from a large number of embedded unsupervised texts. When these embedded features are used by a multi-layer perceptron, long and short-term memory, and other deep neural network technologies, there have also been great improvements in word sense disambiguation (WSD) performance. Su et al. [15] proposed an NMT framework with asynchronous bidirectional decoding. This method adopts the combination of encoder, reverse decoder, and forward decoder that embed the input source statement into bidirectional-hidden states to achieve better translation results.

In 2017, Chinea-Rios et al. [16] proposed the discriminative ridge regression algorithm. This method uses the N-best hypothesis list given by all hypotheses to configure a weight vector so that each sentence is evaluated by professional translators after the output of the editing system. Around the same time, Bahdanau, Cho, and Bengio put forward a collinear model. In a sentence with the same meaning and different languages, corresponding word pairs are more likely to appear in the translation. This model calculates the number of times when two words appear at the same time and then represents different collinear models by calculating the collinearity of different words [17].

Using the model framework of Seq2Seq, Bapna et al. [18] improved the attention model and proposed deep neural machine translation models with transparent attention. The extension of attention mechanism is similar to creating a weighted residual connection along the depth of encoder, allowing simultaneous dispersion of error signals in the encoder depth and time. Before the data set training model is decomposed into sub word units, each sentence is marked with a Morse marker to achieve the goal. Earlier researchers also used graph structures for neural translation until Dou et al. [19] proposed a depth representation for neural translation. For the aggregation function, they used tree structures to merge aggregation nodes of the same depth first and then fed the output of this aggregation node back to the trunk as the input of the next subtree, solving the problem of the previous model's inability to retain characteristics. To solve the complexity of deep model and language, He et al. [20] proposed a residual network of residual connections, directly adding the representation of the previous layer to the following layer. Compared with the aggregation function of tree structure, the attention mechanism of residual connection is more suitable for optimizing the complexity of deep translation.

To support languages with large differences in syntax, Socher et al. [21] put forward a novel model called the deep average network. This network has three basic steps. First, associate the embedded vector average value with the input sequence of the tag. Second, pass the average value to one or more feedforward layers. Third, perform linear classification of the representation of the last layer. A higher accuracy of tasks (translation of various sentences or documents) will thus be obtained. This method uses the Hidden Markov Model to simulate the themes of the sentences and the theme transfer in texts to obtain coherence and is the source of our ideas for improving translation contexts using the Hidden Markov Model.

3 Hidden Markov Model

3.1 Translation Model Focusing on Context

In our method, we determine the topic of each sentence in a coherent document, with the document thus described as a sequence of sentence topics. However, topics are interrelated, and topic changes are continuous, similar to a relationship diagram. This topic sequence forms the document coherence chain. Our coherent capture framework for statistical machine translation uses a document coherence chain built using the Hidden Markov Model.

For the review, the Hidden Markov Model is defined as a quintuple: $\lambda = \{x, y, \pi, a, B\}$, abbreviated as $\lambda = \{x, a, B\}$.

X is the state set for N states: $X = \{x_1, x_2, \ldots, x_n\}$.

O is the set of phrase observation symbols for M possible occurrences of each state $O = \{V_1, V_2, \ldots, V_M\}$.

π is the initial state distribution: $\pi = \{\pi_{\mathrm{i}}\}, \pi_{\mathrm{i}} = P\{q_t = x_i\}, 1 \le j \le N$.

A is the state transition probability matrix: $A = \left\{a_{ij}\right\}, a_{ij} = P\left\{q_{i+1} = x_j\right\}, 1 \le i,\ j \le N$.

B is the probability matrix of the observation $B = \left\{b_{j(k)}\right\}$ representing the probability of state:

$$b_{j(k)} = P\left\{O_t = V_k | q_t = x_j\right\}, \quad 1 \le j \le N, 1 \le k \le M$$

For the state set X (see above), the sequence of hidden states $Y = \{y_1, y_2, \ldots, y_M\}$ is our observation sequence.

The question that requires prediction is

$X = \{x_1, x_2, \ldots, x_N\} = \arg\max P(x_1, x_2, \ldots, x_N | y_1, y_2, \ldots, y_M).$

This formula is equivalent to $\arg\max \prod_{i=1}^{N} P(x_i|y_i) P(y_i|y_{i-1})$.

Both the Gaussian mixture model (GMM) and the dynamic neural network (DNN) as shown in Fig. 1 can fit the probability distribution of an observation sequence, with B acting as the observation state probability matrix of the Hidden Markov Model. The arrow from the Hidden Markov Model to the GMM or DNN means that the observation state probability of a state of the Hidden Markov Model is determined by an output node of the GMM or DNN.

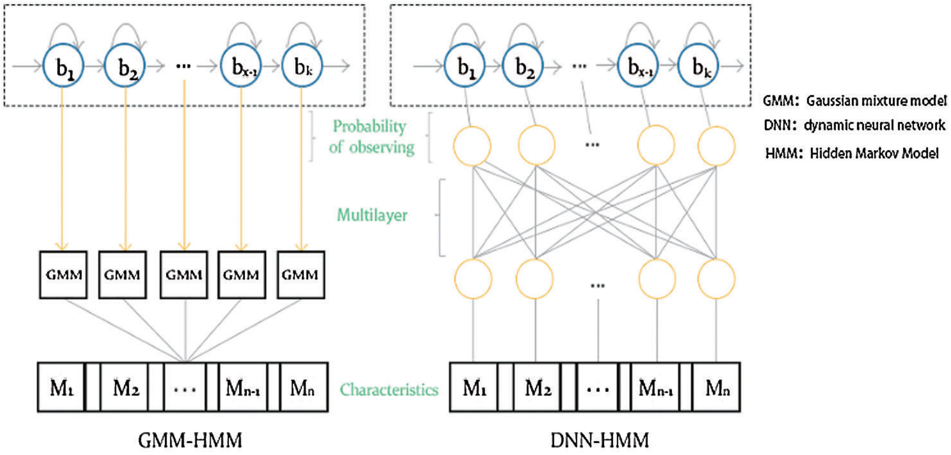


Figure 1: Observation probability matrix

In the Hidden Markov chain, there are many values of state t at any time. Taking the conversion of Pinyin to Chinese characters as an example, for the Pinyin "yike," the possible meanings are one tree, one moment, or one candy. The jth possible value of a state is represented by a symbol. The sequence of states is expanded to obtain a fence net, which is the graph structure for solving the optimal path.

The prediction translation chain of Hidden Markov requires a path in the graph so that the probability value of the corresponding path is the maximum. In the case of Fig. 2, we suppose that the possible value of X at each time is 3, for 3^n combinations. The base number 3 is the width of the fence network, and the index of n is the length of the fence network, so the number of calculations is quite large. We use dynamic programming to solve the probability maximum path, understood as the shortest path of the graph, so that the complexity is proportional to the sequence length. The complexity is O $(n \cdot D \cdot D)$, where n is the length, and D is the width.

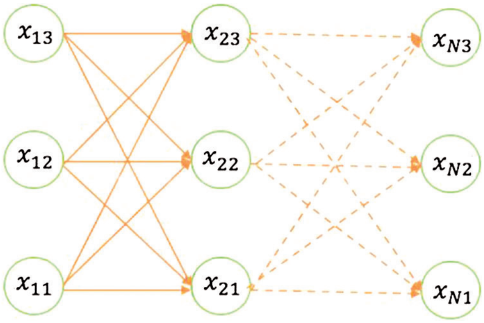


Figure 2: Graph structure of the optimal path

3.2 The Algorithm

Our overall approach implements the following requirements and calculations.

1. In the case of Fig. 2, if the path with the highest probability passes through a specific point of the fence network, the sub path from the starting point to this point must also have the highest probability.
2. Assuming that there are k states at the ith moment, there are k shortest paths from the beginning to the ith moment. The final shortest path must pass through one of them.
3. According to the preceding requirements, when calculating the shortest path of the $(i+1)$th state, only the shortest path from the start to the current k state values and the shortest path from the current state values to the $(i + 1)$th state values need to be considered. For example, the shortest path when t = 3 is equal to the sum of the shortest paths of all state nodes X_{2t} and when t from 2 to the shortest path of each node

In order to record the intermediate variables, two variables δ and ψ are introduced to define the maximum probability value (shortest path) of all single paths with the state i at time t as

$$\delta_t = \max P\left(i_t = i, i_{t-1}, \ldots, i_1, o_t, \ldots, o_1 | \lambda\right), \quad i = 1, 2, \ldots, N, \tag{1}$$

where i_t is the shortest path, O_t is the observation symbol, and λ represents the model parameters. According to this formula, the recurrence formula of variable δ can be obtained as

$$\delta_{t+1}\left(i\right) = \max\left[\delta_j\left(j\right) a_{ji}\right] b_j\left(o_{t+1}\right), \quad i = 1, 2, \ldots, \quad N, t = 1, 2, \ldots, T-1. \tag{2}$$

Among all the single paths $(i_1, i_2, \ldots, i_t)$ defined with state i and time t, the $(T - 1)$th node of the path with the greatest probability is

$$\psi_t\left(i\right) = \arg\max\left[\delta_{t-1}\left(j\right) a_{ji}\right], \quad 1 \leq j \leq N. \tag{3}$$

The input model and observation status are, respectively,

$$\lambda = \{\pi, \mathrm{A}, \mathrm{B}\} \tag{4}$$

The output to find the optimal path is

$$\mathrm{I}^* = \left(i_1^*, i_2^*, \ldots, i_T^*\right) \tag{5}$$

The programmed steps of our method are as follows.

1. Initialize the parameters:

$$\delta_t(i) = \pi_i b_i(o_1), \quad i = 1, 2, \ldots, N \tag{6}$$

$$\psi_t(i) = 0, \quad i = 1, 2, \ldots, N \tag{7}$$

2. According to formulas (13) and (14), we calculate the maximum of the $\delta_t(i)$ and $\psi_t(i)$ when $t = 2, 3, \ldots, T$.

$$\delta_{\mathrm{t}}(i) = \max_{1 \le j \le N} \left[\delta_{t-1}(j)\, a_{ji}\right] b_i(o_i), \quad i = 1, 2, \ldots, N \tag{8}$$

$$\psi_{\mathrm{t}}(i) = \arg\max_{1 \le j \le N} \left[\delta_{t-1}(j)\, a_{ji}\right], \quad i = 1, 2, \ldots, N \tag{9}$$

3. We terminate the calculations when P^* is the maximum of the $\delta\ (i)$:

$$P^* = \max_{1 \le i \le N} \delta_T(i) \tag{10}$$

$$i_T^* = \arg\max_{1 \le i \le N} [\delta_T(i)] \tag{11}$$

$$i_t^* = \Psi_{t+1}\left(i_{t+1}^*\right), \quad t = T-1, \quad T-2, \ldots, 1 \tag{12}$$

This results in the optimal path

$$I^* = \left(i_1^*, i_2^*, \ldots, i_T^*\right). \tag{13}$$

We input the new observation sequence into the HMM and obtain the new sequence, which is the translation result.

3.3 Combination of Algorithm and Translation

We divide the different translation results into different dice. Each die is regarded as a translation result. By training all of the dice in the data set, we obtain the probability of the corresponding results of each die. Our method puts all of the probabilities into a matrix and compares the probabilities of each result using the Viterbi algorithm. We select the die with the highest probability to determine the final translation sentence. We then take all of the translated sentences as different results for the dice and continue applying the Viterbi algorithm again to obtain the translation with the maximum probability according to the calculated probability.

4 Experimental Setup

4.1 Training Database

We use double layer LSTM network to train data, the training data details are in Tab. 1.

As shown in Tab. 1, by training the input layer, embedded layer, and convolution layer, we calculate the related coefficient parameter value.

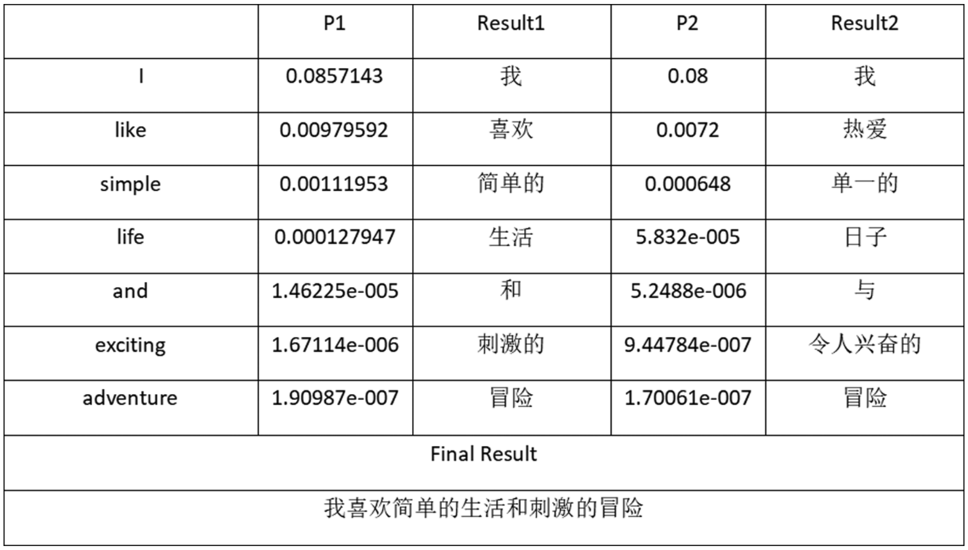
4.2 Test and Results

We align each word and then take the state transition probability matrix and observation state probability matrix as input. Finally, we list the shortest path of each sentence in the article, and form the article sequence. Fig. 3 presents a brief example.

As shown in Fig. 4, we first transform an aligned bilingual parallel sentence pair with a source-side dependency tree into a new dependency-based bilingual linear sequence using word alignment. A unique double CNN then learns the semantic representation of each linear unit at the word level.

Table 1: Epoch training results

Layer (type)	Output shape	Param #	Connected to
input_1 (InputLayer)	(None, none)	0	
input_2 (InputLayer)	(None, none)	0	
embedding_1 (Embedding)	(None, none, 128)	896000	Input_1 [0] [0]
embedding_2 (Embedding)	(None, none, 128)	1280000	Input_2 [0] [0]
cu_dnnlstm_1 (CuDNNLSTM)	(None, none, 256)	395264	embedding_1 [0] [0]
cu_dnnlstm_3 (CuDNNLSTM)	(None, none, 256)	395264	embedding_2 [0] [0] cu_dnnlstm_1 [0] [1] cu_dnnlstm_1 [0] [2]
cu_dnnlstm_2 (CuDNNLSTM)	(None, 256)	526336	cu_dnnlstm_1 [0] [0]
cu_dnnlstm_4 (CuDNNLSTM)	(None, None, 256)	526336	cu_dnnlstm_3 [0] [0] cu_dnnlstm_2 [0] [1] cu_dnnlstm_2 [0] [2]
dense_1 (Dense)	(None, none, 10000)	2570000	cu_dnnlstm_4 [0] [0]



	P1	Result1	P2	Result2
I	0.0857143	我	0.08	我
like	0.00979592	喜欢	0.0072	热爱
simple	0.00111953	简单的	0.000648	单一的
life	0.000127947	生活	5.832e-005	日子
and	1.46225e-005	和	5.2488e-006	与
exciting	1.67114e-006	刺激的	9.44784e-007	令人兴奋的
adventure	1.90987e-007	冒险	1.70061e-007	冒险
Final Result				
我喜欢简单的生活和刺激的冒险				

Figure 3: Sample translation probabilities and results

For word alignment, the forward and backtracking algorithms adopt one-to-many or one-to-one methods, subsection and retain context dependency through the aligning process. However, the Hidden Markov Model does not require one-to-many alignment. When we start translating from the second word, we deduce the optimal solution of the second word using the maximum probability of the first word. Further, when using HMM to segment sentences, the optimal solution of each sentence is integrated into a new text is done to form an article sequence. The simplification is possible because the Hidden Markov Model does not need to store the context dependency. We can directly use the article sequence as a new variable to execute another Hidden Markov Model operation.

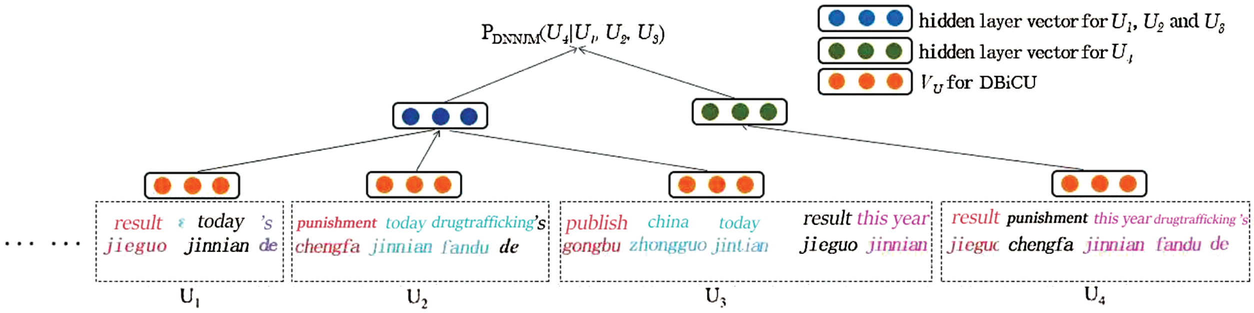

$P_{DNNJM}(U_4|U_1, U_2, U_3)$
hidden layer vector for U_1, U_2 and U_3
hidden layer vector for U_4
V_U for DBiCU
... ...
result today 's
jieguo jinnian de
punishment today drugtrafficking's
chengfa jinnian fandu de
publish china today result this year
gongbu zhongguo jintian jieguo jinnian
result punishment this year drugtrafficking 's
jieguc chengfa jinnian fandu de
U1
U2
U3
U4

Figure 4: Context-based statistical machine translation

Second, we use the HMM chain (see Fig. 5) to find the optimal path, carrying out both forward and backtracking at the same time. Doing so results in the accuracy of the results being much higher than those from the gradient descent method.

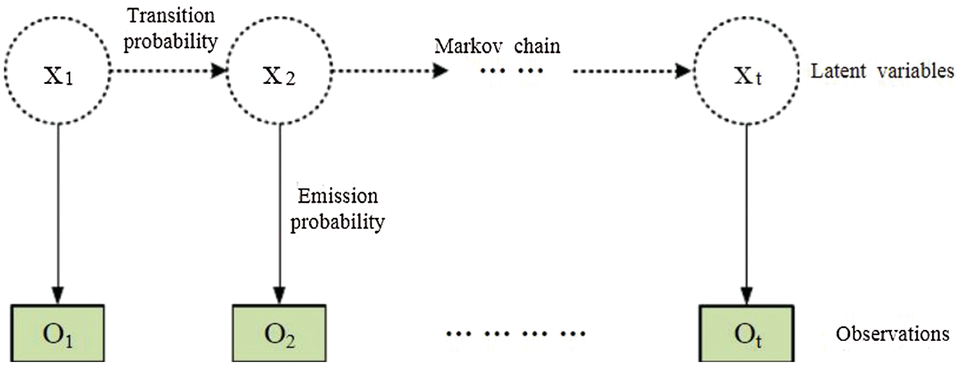


Figure 5: Hidden Markov model chain

In addition, Fig. 5 depicts a theme-based coherence model for document-level machine translation. The consistency chain of the source document is generated by the Markov topic model, and the consistency chain of the source document is projected to the corresponding target document using the MaxEnt prediction model. The projected coherence chain captures the subject-related constraints of word or phrase selection in target document translation.

According to the experimental data, the accuracy of using the entire sentence when constructing context is much higher than using only the probabilities associated with each topic word, as shown in Tab. 2. Although word-based probabilities are representative in some situations, overall accuracy of words is not very high.

Table 2: Probability of topic words selection

Word	P	Word	P
United	0.0209182	Russia	0.00637757
States	0.0203053	Security	0.00617798
China	0.00922345	International	0.00601291
Countries	0.00842481	...	...
Military	0.00749308	Action	0.000886684
Defense	0.00702691	...	...
Bush	0.00658136	Movement	0.000151846

5 Conclusion

In this study, we explored context-based processing for machine language translation. We used a Hidden Markov Model to decompose target sentences to identify possible translation paths. Through forward and backward tracking, our model calculates the probability of each translation result to form the article sequence. Each sentence is then taken as a translation unit, with consideration of mutual influence between sentences. The Hidden Markov Model calculates the maximum probabilities to determine the best contextual results. However, the experiments reveal many deficiencies, such as the small number of datasets stored in the database. We plan to increase the size of our datasets in future work to improve this performance. With the continuing growth of the language corpus, we expect the meaning of machine-translated sentences to move closer to the original meaning.

Funding Statement: The financial support provided from the Cooperative Education Fund of China Ministry of Education (201702113002 and 201801193119), Hunan Natural Science Foundation (2018JJ2138) and Degree and Graduate Education Reform Project of Hunan Province (JG2018B096) are greatly appreciated by the authors.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] W. Liang, M. Tang, J. Long, X. Peng, J. Xu, K. C. Li, "A secure fabric blockchain based data transmission technique for industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3582–3592, 2019.

[2] W. Liang, W. H. Huang, J. Long, K. C. Li and D. F. Zhang, "Deep reinforcement learning for resource protection and real-time detection in IoT environment," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6392–6401, 2020.

[3] B. Zhang, D. Xiong, J. Su and J. Luo, "Future-aware knowledge distillation for neural machine translation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2278–2287, 2019.

[4] Z. W. Feng, "Parallel development of machine translation and artificial intelligence," *Foreign Languages (Journal of Shanghai Foreign Studies University)*, vol. 41, no. 6, pp. 35–48, 2018.

[5] Z. H. Wang, "Research and development of machine translation technology," *Electronic Production*, no. 22, pp. 64–66, 2018.

[6] D. Bahdanau, K. Cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, San Diego, CA: United States, 2015.

[7] Y. Yang, *Research and Implementation of Uygur Chinese Translation Based on Neural Network*. Sichuan, China: University of Electronic Science and Technology, 2019.

[8] Y. Shi, Y. Wang and S. Q. Wu, "Machine translation system based on self attention model," *Computer and Modernization*, no. 7, pp. 9–14, 2019.

[9] P. Martínez-Gómez, G. Sanchis-Trilles and F. Casacuberta, "Online adaptation strategies for statistical machine translation in post-editing scenarios," *Best Papers of Iberian Conf. on Pattern Recognition and Image Analysis*, vol. 45, no. 9, pp. 3193–3203, 2012.

[10] K. Chen, R. Wang, M. Utiyama, E. Sumita and T. Zhao, "Neural machine translation with sentence-level topic context," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 1970–1984, 2019.

[11] Y. Liu, K. Wang, C. Q. Zong and K. Y. Su, "A unified framework and models for integrating translation memory into phrase-based statistical machine translation," *Computer Speech & Language*, vol. 54, pp. 176–206, 2019.

[12] M. Narouel, M. Ahmadi and A. Sami, "Word sense disambiguation by sequential patterns in sentences," *Natural Language Engineering*, vol. 21, no. 2, pp. 251–269, 2015.

[13] Á. Peris and F. Casacuberta, "Online learning for effort reduction in interactive neural machine translation," *Computer Speech and Language*, vol. 58, pp. 98–126, 2019.

[14] D. Bahdanau, K. Cho and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. ICLR2015*, San Diego, CA, United States, 2015.

[15] J. Su, X. Zhang, Q. Lin, Y. Qin, J. Yao *et al.,* "Exploiting reverse target-side contexts for neural machine translation via asynchronous bidirectional decoding," *Artificial Intelligence*, vol. 277, pp. 103168, 2019.

[16] M. Chinea-Rios, F. Casacuberta and G. Sanchis-Trilles, "Discriminative ridge regression algorithm for adaptation in statistical machine translation," *Pattern Analysis and Applications*, vol. 22, no. 4, pp. 1293–1305, 2019.

[17] H. H. He, "The parallel corpus for information extraction based on natural language processing and machine translation," *Expert Systems*, vol. 36, no. 5, pp. 131, 2018.

[18] A. Bapna, M. X. Chen, O. Firat, Y. Cao and Y. Wu, "Training deeper neural machine translation models with transparent attention. in *Proc. EMNLP*, Brussels, Belgium, 2018.

[19] Z. Y. Dou, Z. P. Tu, X. Wang, S. M. Shi and T. Zhang, "Exploiting deep representations for neural machine translation," in *Proc. EMNLP*, Brussels, Belgium, 2018.

[20] Y. Y. Shen, X. Tan, D. He, T. Qin and T. Y. Liu, "Dense information flow for neural machine translation," in *NAACL HLT*, New Orleans, LA, United States, 2018.

[21] M. Iyyer, J. Boyd-Graber, L. Claudino, R. Socher and H. Daume, "A neural network for factoid question answering over paragraphs," in *Proc. EMNLP*, Doha, Qatar, 2014.