Annals of Emerging Technologies in Computing (AETiC)

Vol. 6, No. 3, 2022

Research Article

Analysis of Intelligent English Chunk Recognition based on Knowledge Corpus

Mei Zhang

Lanzhou City University, Lanzhou, Gansu 730070, China
zhangmei@edu.cn

Received: 20th April 2022; Accepted: 19 June 2022; Published: 1st July 2022

Abstract: Chunks play an important role in applied linguistics, such as Teaching English as a Second Language (TESL) and Computer-Aided Translation (CAT). Although corpora have already been widely used in the areas mentioned above, annotation and recognition of chunks are mainly done manually. Computer- and linguistic-based chunk recognition is significant in natural language processing (NLP). This paper briefly introduced the intelligent recognition of English chunks and applied the Recurrent Neural Network (RNN) to recognise chunks. To strengthen the RNN, it was improved by Long Short Term Memory (LSTM) for recognising English chunk. The LSTM-RNN was compared with support vector machine (SVM) and RNN in simulation experiments. The results suggested that the performance of the LSTM-RNN was always the highest when dealing with English texts, no matter whether it was trained using a general corpus or a corpus of specialised domain knowledge.

Keywords: ***applied linguistics; chunk recognition; corpus; recurrent neural network***

1. Introduction

Much of the progress in the development of linguistics has been the result of paradigm shifts within the discipline. Moreover, computer science and computational linguistics have helped form an interdisciplinary field that has made a considerable contribution to how humans can better understand language. For example, the emergence of the corpus is an epochal revolution. However, in many studies based on corpora, manual intervention is required. For instance, almost all corpus-based language chunk studies currently use manual annotation, which significantly limits the research and reapplication of language blocks. Computers cannot directly understand natural language, as would be requited in applications such as language translation and speech recognition [1]. Hence, the need for algorithms to convert the natural language into the computer language.

These algorithms need to address such issues as the same word in different grammars having different meanings [2]. The computer cannot rigidly recognise individual words but needs to recognise chunks, i.e., it divides a sentence into different syntactic modules [3], and the subsequent processing of natural language is based on the types of chunks. For example, Seung-Hoon *et al.* [4] put forward a phrase-based lexical analysis model for Korean and found from the experimental results that the phrase-based model performed better than the morpheme-based model. Sarkar *et al.* [5] proposed an effective method to segment and label pass-phrase utterances. Moreover, Christie *et al.* [6] proposed a means for simultaneous semantic segmentation and prepositional phrase attachment resolution. Napolitano *et al.* [7] introduced some techniques that could preprocess breast cancer pathology reports to facilitate the extraction of cancer stage-related chunks, one using the free software RapidMiner and the other using the K-nearest algorithm to construct a layout classifier. They found that the layout classifier had an accuracy of 99.4%. A new dynamic chunking method proposed by Lin *et al.* [8] mapped original sequences into a fixed number of chunks. The experiment based on three databases found that this method improved

Mei Zhang, "Analysis of Intelligent English Chunk Recognition based on Knowledge Corpus", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 11-18, Vol. 6, No. 3, 1st July 2022, Published by International Association for Educators and Researchers (IAER), DOI: 10.33166/AETiC.2022.03.002, Available: http://aetic.theiaer.org/archive/v6/v6n3/p2.html.

recognition accuracy, robustness, and recognition efficiency. Kai *et al.* [9] used a temporal back-propagation algorithm based on context-sensitive chunks to segment the text into chunks with appended contextual observations. Guo *et al.* [10] proposed a framework combining bidirectional long short-term memory (LSTM) with the convolutional neural network to recognise implicit discourserelations and found through experiments that the method could effectively recognise implicit discourse relations. Our paper used LSTM-Recurrent Neural Network (RNN) to automatically recognise chunks in the corpus, which will positively affect related research in language teaching, machine translation, and other applied fields of linguistics. Compared to previous studies, our study recognised English chunks by taking the RNN's advantage for serial information and highlighted the important information with LSTM to improve the recognition performance of the RNN, which are the novelties of this paper.

2. Intelligent Recognition of English Chunks

2.1. The Basic Process of Chunk Recognition

When processing a natural language text, such as in information retrieval and machine translation applications, the computer cannot process it word by word because words will have different meanings in different grammatical environments. Take machine translation of English as an example; when English words can be both nouns and verbs, their specific meanings are determined according to the grammatical environment of the sentence they are in [11]. Therefore, before machine translation of an English text, text needs to be divided into chucks through a classification process and labelled according to their type.

Chunk recognition is also called shallow syntactic parsing. The purpose of chunk recognition is to segment a sentence into different syntactic modules that can also be used as features needed for the computer processing of natural language. Chunk recognition can be described as a labelling task that first determines the category of the current word, then confirms the category boundary and ensures that the words within the boundary belong to the same category [12]. In this paper, the IOBES labelling strategy is adopted: the word that is determined currently is represented by X, B-X denotes that the current word is the initial word of the segmented chunk, I-X indicates that the current word is the middle word of the segmented chunk, E-X indicates that the current word is the ending word of the segmented chunk, S-X denotes that the current word is the word splitting of the segmented chunk, and O-X means that the current word does not belong to any segmented chunk.

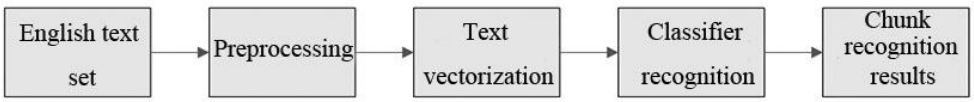

Figure 1. The basic flow of chunk recognition

The basic process flow for recognising English chunks is shown in Figure 1. First, English texts are collected and divided into chunks in the form of manual annotation, and the types of chunks are labelled to build a corpus [13]. Then, the text is preprocessed through removing irrelevant words and punctuation, to reduce the interference of redundant words to the recognition. The text is then vectorised using the single thermal representation method [14], where the single thermal representation requires the use of a word list. The number of words in the word list determines the dimension of the text vector, and every word occupies one dimension. When the word to be vectorised is contained the word list, the dimension where the word is located takes the value of 1, and the rest of the dimensions take the value of 0. Finally, the vectorised text is recognised and labelled by the classifier. Commonly used classifiers include SVMs, decision trees, neural networks, etc [15].

In the process of chunk recognition, besides the classifier, the most important thing is the collection of English texts used for training the classifier, i.e., the manually annotated corpus collection. Usually, an open general corpus is used as the training set to ensure the generality of the trained chunk recognition classifier and the convenience of training. The advantage of using an open general corpus is that it does not require additional data collection and annotation [16]. Still, the disadvantage is that it is inclined to include common English language materials. Though the application of a general corpus ensures the generality of the classifier, it is difficult to guarantee the recognition effect when facing the specialised knowledge domain, i.e., English texts containing many proper nouns. Therefore, based on the general

corpus, this paper additionally collected journal papers from different scientific research fields and reconstructed the corpus containing specialised knowledge [17].

2.2. The Improved RNN Classifier for Chunk Recognition

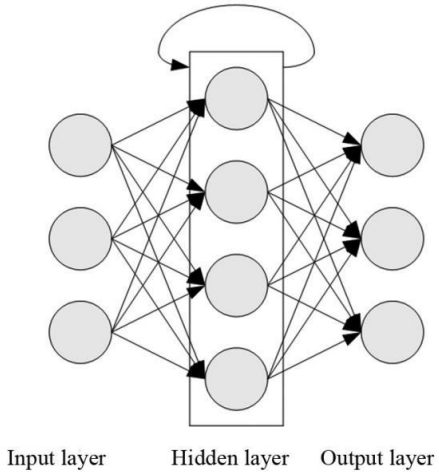

Figure 2. The basic structure of an RNN

The basic principle of chunk recognition annotation for English is classifying and recognising a text using a classifier after the text has been transformed into a vector [18]. For natural languages, the meaning of a text and the kinds of chunks composed of words is affected by the context. SVMs, decision trees, and traditional back-propagation neural networks (BPNNs) do not consider the context of a text when being used as classifiers, i.e., the text vectors are independent of each other.

Like a BPNN, an RNN is a structure containing input, hidden, and output layers. However, there is only one hidden layer in the RNN, which is the difference between a BPNN and an RNN. The single hidden layer has a self-connected loop, enabling the neural network to use the current input information and the historical information in the training and use process. In short, the input sequence of information will affect the training and use of classifiers, which fits quite well with the chunk recognition and labelling of English texts. In the process of using the RNN, the vectorised sentences are input to the input layer nodes of the RNN according to the word order, and then the data in the input layer nodes are input to the hidden layer nodes. A self-connected loop is between the hidden layer nodes of the RNN, which means that the data obtained from calculating the nodes in the hidden layer are calculated again in addition to calculating the data from the input layer nodes. The calculation formula in the hidden layer first linearly fits the data with the linear formula and then nonlinearly fits the linear data with the nonlinear activation function to obtain the nonlinear law between data. The final calculated results are output from the output layer nodes.

The forward calculation formula for the hidden layer in the RNN is:

$$\begin{cases} b_h^t = f\left(\sum_{i=1}^{I} \omega_{ih} x_i^t + \sum_{h'=1}^{H} \omega_{hh'} b_{h'}^{t-1}\right) \\ a_k^t = \sum_{h=1}^{H} \omega_{hk} b_h^t \end{cases} \tag{1}$$

where I means the number of the input layer nodes, H means the number of the hidden layer nodes, x_i^t denotes the input of sequence t in the input node i, ω_{ih} denotes the weight between the input node i and hidden node h, $\omega_{hh'}$ represents the weight between the hidden node h and another hidden node h', b_h^t is the output of sequence t in hidden node h, $f(\cdot)$ is the activation function, a_k^t is the output of sequence t in output node k, and ω_{hk} is the weight between the hidden node h and output node k. After the results are obtained using the forward calculation of Equation (1), they are compared with the actual labels in the training samples. Since this study focuses on the chunk labelling of English texts, the error is calculated by cross-entropy [19]. Then, the weights in the forward calculation formula are reversely adjusted until the error converges to stability or the training reaches a set number of times.

The context can be effectively utilised when classifying and recognising text chunks with the loop structure of the hidden layer in RNN. Theoretically, RNN can process the information of the whole sequence. However, in practical use, as the input information circulates in the loop of the hidden layer, its influence in the hidden nodes gradually decreases, and it cannot effectively process long sentences. To solve the shortcoming mentioned above, this paper improves RNN with LSTM. The computational flow of the improved LSTM-RNN is as follows.

① Texts in the training samples are preprocessed and vectorised using the single thermal representation method.

② The vectorised texts are input into the LSTM, and the forward calculation formula within the LSTM is:

$$\begin{cases} f_t = \sigma(b_f + U_f x_t + W_f h_{t-1}) \\ s_t = f_t s_{t-1} + g_t \sigma(b + U x_t + W h_{t-1}) \\ g_t = \sigma(b_g + U_g x_t + W_g h_{t-1}) \\ h_t = \tanh(s_t) q_t \\ q_t = \sigma(b_q + U_q x_t + W_q h_{t-1}) \end{cases} \tag{2}$$

where f_t stands for the output of the forgetting gate [20], s_t stands for the output of the cyclic gate, g_t is the external input gate unit, q_t is the output gate unit, and b, U, and W stand for the bias term, input term weight and cyclic gate weight of the corresponding gate, respectively.

③ The output gate data obtained from the LSTM calculation is used as the input data in Equation (1) for forward calculation. The error is calculated by comparing the calculation results with the actual labels of the training samples, which is used to reversely adjust the weight parameters in the LSTM and RNN. The error is still calculated using cross-entropy, and the calculation formula is:

$$E = -\frac{\sum_{i}\sum_{c=1}^{M} y_{ic} \log(p_{ic})}{N} \tag{3}$$

where y_{ic} stands for a symbolic function of whether the sample i belongs to class c (its value is 1 if it belongs to class c; otherwise, its value is 0), M is the total number of classes, N represents the sample size, and p_{ic} stands for the probability that sample i is a member of class c.

The above procedures are repeated until the error converges to the preset threshold or the iteration reaches the preset maximum number.

3. Simulation Recognition

3.1. Experimental Data

The general corpus used for the experiments is from the Brigham Young University corpus (https://corpus.byu.edu/), one of the most widely used online corpora and supported by hundreds of universities, and thousands of individuals, and organisations. The online corpus contains a wide variety of corpora, involving news, Wikipedia, film and television works. Training and testing samples were selected from the News on the Web (NOW) corpus, containing more than 14 billion words and 20 national languages. Ten thousand simple sentences in English were selected as the data samples for the experiment.

In addition, the top 100,000 most frequently occurring words were selected from the English part of the NOW corpus as the word list for the single thermal representation method, and the words outside this word list were represented by $\langle UNK \rangle$.

A general corpus was established by randomly selecting 7000 single sentences from the experimental data samples for training the algorithm, and the remaining 3000 single sentences were used for testing. In addition to the general corpus, to further improve the recognition performance of the English chunk intelligent recognition algorithm for texts in specialised fields, the research reported here also made use of the latest issues of 20 authoritative journals in fields such as mathematics and chemistry, materials science, aviation, agriculture, and new energy and selected ten papers from each journal, i.e., a total of 200 papers.

The main bodies of 140 papers were selected and added to the general corpus to build a knowledge corpus, and the main bodies of the remaining 60 papers were added to the test samples.

3.2. Experimental Setup

English text chunks were recognised by LSTM-RNN. The relevant parameters are as follows. LSTM had two hidden layers, each contained 1024 nodes, and sigmoid was used as the activation function; RNN has two hidden layers, each contained 1024 nodes, and the activation function was the same; 1000 iterations was the maximum, and the learning rate was set as 0.1.

To verify the performance of LSTM-RNN for chunk recognition, it was compared with two algorithms, SVM and RNN. The kernel function of SVM was sigmoid, and the penalty parameter was 1. The parameters of RNN were the same as the RNN part in LSTM-RNN.

3.3. Evaluation Criteria

The recognition performance of the algorithm was evaluated by precision and recall rate, and the calculation formulas are:

$$\begin{cases} P = \dfrac{x}{n} \times 100\% \\ R = \dfrac{x}{m} \times 100\% \end{cases} \tag{4}$$

where P is the precision, R is the recall rate, x is the number of samples that are correctly recognised, n is the number of samples recognised by the algorithm, and m is the number of standard answers in the test sample.

3.4. Experimental Results

Figure 3 illustrates the performance test results of the three chunk recognition algorithms for recognition of the general test samples and knowledge test samples after training with the general corpus. The precision and recall rate of SVM for recognising the general test samples were 82.3% and 81.2%, those of RNN were 90.1% and 89.6%, and those of LSTM-RNN were 97.8% and 96.7%. Facing the knowledge test samples, the precision and recall rate of SVM were 76.8% and 75.7%, those of RNN were 85.4% and 84.6%, and those of LSTM-RNN were 90.2% and 89.8%. It was seen from Figure 3 that when recognising the general test sample and the knowledge test sample, LSTM-RNN had the best recognition performance, followed by RNN and SVM. In addition, the performance of the three chunk recognition algorithms trained by the general corpus was lower in recognising the knowledge test samples than the general test samples.

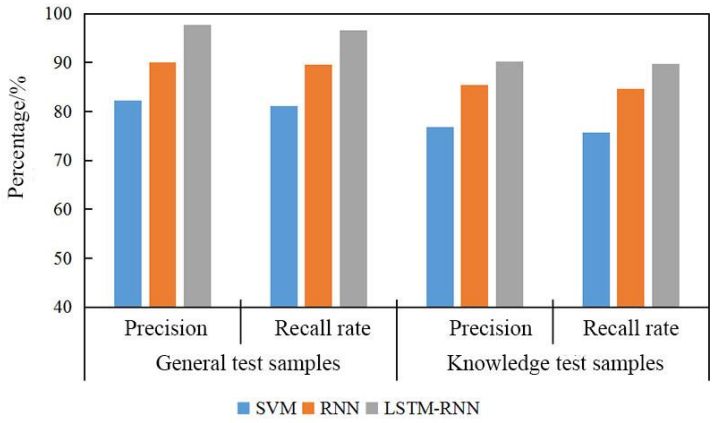

Figure 3. Recognition performance of three algorithms for two kinds of test samples after training by the general corpus

Figure 4 illustrates the performance test results of the three chunk recognition algorithms for recognition of the general test samples and knowledge test samples after training with the knowledge

corpus. The precision and recall rate of SVM for recognising the general test samples were 82.4% and 81.4%, those of RNN were 90.1% and 89.8%, and those of LSTM-RNN were 97.7% and 96.7%. When recognising the knowledge test samples, the precision and recall rate of SVM were 79.4% and 79.7%, those of RNN were 88.3% and 87.7%, and those of LSTM-RNN were 97.1% and 96.6%. It was observed from Figure 4 that LSTM-RNN performed the best when recognising the two types of test samples; among the three chunk recognition algorithms trained by the knowledge corpus, the performance of SVM and RNN was lower in recognising the knowledge test samples than the general test samples. The performance reduction of LSTM-RNN was not as significant as SVM and RNN.

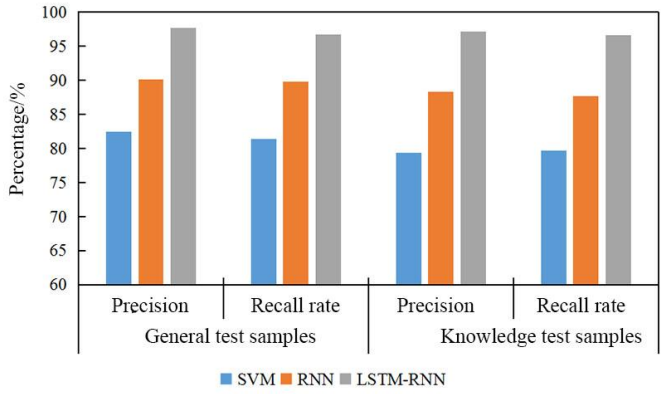

Figure 4. Recognition performance of three algorithms for two kinds of test samples after training by the knowledge corpus

The average time taken by the three algorithms to recognise chunks of English texts is shown in Figure 5, 2.4 s for SVM, 1.9 s for RNN, and 1.1 s for LSTM-RNN. Figure 5 showed that LSTM-RNN was the most efficient, followed by RNN and SVM.

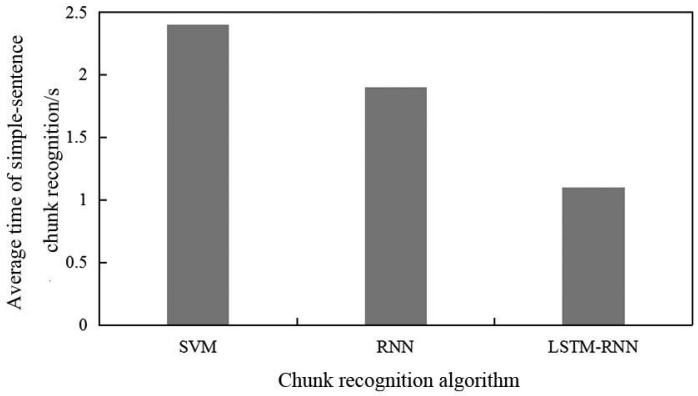

Figure 5. Average time of three algorithms for single-sentence chunk recognition

4. Discussion

Natural language recognition is an important part of computer intelligence. Accurate natural language recognition can significantly enhance the interactivity between computers and humans. In addition, natural language recognition is also crucial when retrieving textual information or performing machine translation. Chunk division and recognition of sentences facilitate the accurate recognition of natural language. Since the order of words in natural language affects the expression of word meanings, this paper adopted an RNN that can use historical input information to perform chunk recognition on English texts and applied LSTM to compensate for the loss of historical information during the use of RNN. Finally, simulation experiments were conducted on LSTM-RNN, and it was compared with SVM and RNN to verify the recognition performance of LSTM-RNN. The results have shown above. Whether it was trained by the general corpus or the knowledge corpus, LSTM-RNN always had the best recognition performance when facing the test sample set, followed by RNN and SVM. The reasons are as follows. SVM was a linear classifier, so it could not effectively fit the nonlinear classification law even if the input information was mapped to a high-dimensional space using the kernel function.

Moreover, SVM only considered the current input data when classifying data and could not utilise the contextual information; therefore, its performance was the worst. RNN could effectively utilise the contextual information, and the activation function in the neural network could effectively fit the nonlinear classification law; therefore, its chunk recognition performance was superior to SVM. LSTM-RNN inherited the advantages of RNN. The use of LSTM made up for the loss of historical information during the use of RNN, improving the performance in recognising long sentences; therefore, its chunk recognition performance was the best.

After training with the knowledge corpus, the recognition performance of the three chunk recognition algorithms improved when facing the knowledge test samples. The knowledge test samples contained texts from journal papers, which included many specialised words, and the types of chunks were different from those in common language texts. Therefore, the chunk recognition algorithms trained with the general corpus treated specialised words as common words, resulting in lower recognition accuracy. The knowledge corpus was added with specialised knowledge texts from journal papers, supplementing the relevant classification laws, improving the recognition accuracy.

5. Conclusion

This paper briefly introduced the intelligent recognition of English chunks, used RNN for chunk recognition, improved RNN with LSTM, and compared LSTM-RNN with SVM and RNN in simulation experiments. The results are as follows. Whether trained by the general corpus or the knowledge corpus, when facing the test sample set, LSTM-RNN always had the best recognition performance, followed by RNN and SVM. After being trained with the knowledge corpus, the recognition performance of all three chunk recognition algorithms for knowledge test samples improved. LSTM-RNN had the highest efficiency in recognising English chunks, followed by RNN and SVM.

References

[1] Suchismita Maiti, Utpal Garain, Arnab Dhar and Sankar De, "A novel method for performance evaluation of text chunking", *Language Resources & Evaluation*, Print ISSN: 1574-020X, Online ISSN: 1574-0218, pp. 215-226, Vol. 49, No. 1, 13rd August 2015, Published by Springer Nature B.V., DOI: 10.1007/s10579-013-9250-3, Available: https://link.springer.com/article/10.1007/s10579-013-9250-3.

[2] Almira Fiana Dhara and Rully Agus Hendrawan, "Rancang Bangun Ekstraksi Ekspresi Kata Kerja pada Ulasan Pelanggan Dengan Text Chunking untuk Memaparkan Pengalaman Penggunaan Produk", *Jurnal Teknik ITS*, Online ISSN: 2337-3539, Vol. 6, No. 2, September 2017, DOI: 10.12962/j23373539.v6i2.23151, Available: http://ejurnal.its.ac.id/index.php/teknik/article/view/23151.

[3] Juan Luo and Yves Lepage, "Extraction of Potentially Useful Phrase Pairs for Statistical Machine Translation", *Journal of Information Processing*, Online ISSN: 1882-6652, pp. 344-352, Vol. 23, No. 3, May 2015, Published by Information Processing Society of Japan Production services SANBI Printing Co. Ltd, DOI: 10.2197/ipsjjip.23.344, Available: https://www.jstage.jst.go.jp/article/ipsjjip/23/3/23_344/_article.

[4] Seung-Hoon NA and Young-Kil KIM, "Phrase-Based Statistical Model for Korean Morpheme Segmentation and POS Tagging", *IEICE Transactions on Information & Systems*, Print ISSN: 0916-8532, Online ISSN: 1745-1361, pp. 512-522, Vol. 101, No. 2, 1st February 2018, Published by J-stage, DOI: 10.1587/transinf.2017EDP7085, Available: https://www.jstage.jst.go.jp/article/transinf/E101.D/2/E101.D_2017EDP7085/_article.

[5] Achintya Sarkar and Zheng-Hua Ta, "Self-Segmentation of Pass-Phrase Utterances for Deep Feature Learning in Text-Dependent Speaker Verification", *Computer Speech & Language*, Online ISSN:0885-2308, Vol. 70, No. 6, November 2021, Published by Elsevier, DOI: 10.1016/j.csl.2021.101229, Available: https://www.sciencedirect.com/science/article/abs/pii/S088523082100036X.

[6] Gordon Christie, Ankit Laddha, Aishwarya Agrawal, Stanislaw Antol, Yash Goyal *et al*., "Resolving vision and language ambiguities together: Joint segmentation & prepositional attachment resolution in captioned scenes", *Computer Vision and Image Understanding*, Online ISSN: 1090-235X, pp. 101-112, Vol. 163, October 2017, DOI: 10.1016/j.cviu.2017.09.001, Available: https://www.sciencedirect.com/science/article/abs/pii/S1077314217301571.

[7] Bin Gu, Xin Quan, Yunhua Gu and Victor Sheng, "Chunk incremental learning for cost-sensitive hinge loss support vector machine", *Pattern Recognition*, Online ISSN: 0031-3203, pp. 196-208, Vol. 83, November 2018, Published by Elsevier, DOI: 10.1016/j.patcog.2018.05.023, Available: https://www.sciencedirect.com/science/article/abs/pii/S0031320318301973.

[8] Giulio Napolitano, Adele H Marshall, Peter Hamilton and Anna T. Gavin, "Machine learning classification of surgical pathology reports and chunk recognition for information extraction noise reduction", *Artificial Intelligence in Medicine*, pp. 77-83, Vol. 70, 8th June 2016, Published by Elsevier, DOI: 10.1016/j.artmed.2016.06.001, Available: https://www.sciencedirect.com/science/article/pii/S0933365716302275.

[9] Wei-Cheng Lin and Carlos Busso, "Chunk-Level Speech Emotion Recognition: A General Framework of Sequence-to-One Dynamic Temporal Modeling", *IEEE Transactions on Affective Computing*, Online ISSN: 1949-3045, pp. 1-14, May 2021, Published by Institute of Electrical and Electronics Engineers, DOI: 10.1109/TAFFC.2021.3083821, Available: https://ieeexplore.ieee.org/document/9442335.

[10] Kai Chen and Qiang Huo, "Training Deep Bidirectional LSTM Acoustic Model for LVCSR by a Context-Sensitive-Chunk BPTT Approach", in *Proceedings of Interspeech 2015*, 6th September 2015, Dresden, Germany, DOI: 10.21437/Interspeech.2015-714, pp. 3600-3604, Available: https://www.isca-speech.org/archive/interspeech_2015/chen15r_interspeech.html.

[11] Fengyu Guo, Ruifang He and Jianwu Dang, "Implicit Discourse Relation Recognition via a BiLSTM-CNN Architecture with Dynamic Chunk-based Max Pooling", *IEEE Access*, Online ISSN: 2169-3536, pp. 169281-169292, Vol. 7, November 2019, Published by IEEE, DOI: 10.1109/ACCESS.2019.2954988, Available: https://ieeexplore.ieee.org/document/8908659.

[12] Bing Li, Xiaochun Yang, Rui Zhou, Bin Wang, Chengfei Liu *et al*., "An Efficient Method for High Quality and Cohesive Topical Phrase Mining", *IEEE Transactions on Knowledge & Data Engineering*, Print ISSN: 1041-4347, Online ISSN: 1558-2191, pp. 120-137, Vol. 31, No. 1, 6th April 2018, Published by IEEE, DOI: 10.1109/TKDE.2018.2823758, Available: https://ieeexplore.ieee.org/document/8332520.

[13] Duyu Tang, Bing Qin, Furu Wei, Li Dong, Ting Liu *et al*., "A Joint Segmentation and Classification Framework for Sentence Level Sentiment Classification", *IEEE/ACM Transactions on Audio Speech & Language Processing*, Print ISSN: 2329-9290, Online ISSN: 2329-9304, pp. 1750-1761, Vol. 23, No. 11, 29th June 2015, Published by IEEE, DOI: 10.1109/TASLP.2015.2449071, Available: https://ieeexplore.ieee.org/document/7138591.

[14] Susana Silva, Carolina Magalhães Dias and São Luís Castro, "Domain-Specific Expectations in Music Segmentation", *Brain Sciences*, Online ISSN: 2076-3425, pp. 169-, Vol. 9, No. 7, 17th July 2019, Published by MDPI, DOI: 10.3390/brainsci9070169, Available: https://www.mdpi.com/2076-3425/9/7/169.

[15] Dennis Norris, Kristjan Kalm, Jane Hall, "Chunking and redintegration in verbal short-term memory", *Journal of Experimental Psychology Learning Memory and Cognition*, Print ISSN: 0278-7393, Online ISSN: 1939-1285, pp. 872-893, Vol. 45, No. 5, May 2020, Published by American Psychological Association, DOI: 10.1037/xlm0000762, Available: https://doi.apa.org/fulltext/2019-58802-001.html.

[16] Kevin B Clark, "Natural chunk-and-pass language processing: Just another joint source-channel coding model?", *Communicative & Integrative Biology*, Online ISSN: 1942-0889, pp. 1-2, Vol. 11, No. 2, 30 September 2020, Published by Austin, TX: Landes Bioscience, DOI: 10.1080/19420889.2018.1445899, Available: https://www.tandfonline.com/doi/full/10.1080/19420889.2018.1445899.

[17] Yiwen Mo, Bo Chen, Pei Lei, "Boundary Recognition of Light-Pause Marks via Grammar Testing Method", *Wuhan University Journal of Natural Sciences*, Print ISSN: 1007-1202, Online ISSN: 1993-4998, pp. 230-236, Vol. 23, No. 03, 17 May 2018, Published by Springer Nature, DOI: 10.1007/s11859-018-1315-0, Available: https://link.springer.com/article/10.1007/s11859-018-1315-0.

[18] Xuejian Rong, Chucai Yi and Yingli Tian, "Unambiguous Scene Text Segmentation With Referring Expression Comprehension", *IEEE Transactions on Image Processing*, Print ISSN: 1057-7149, Online ISSN: 1941-0042, pp. 591-601, Vol. 29, 26th July 2019, Published by IEEE, DOI: 10.1109/TIP.2019.2930176, Available: https://ieeexplore.ieee.org/document/8777293.

[19] Chengliang Li, Aixin Sun, Jianshu Weng and Qi He, "Tweet Segmentation and Its Application to Named Entity Recognition", *IEEE Transactions on Knowledge & Data Engineering*, Print ISSN: 1041-4347, Online ISSN: 1558-2191, pp. 558-570, Vol. 27, No. 2, 30th May 2015, Published by IEEE, DOI: 10.1109/TKDE.2014.2327042, Available: https://ieeexplore.ieee.org/document/6823714.

[20] Shabistan Ruhi Ansari and Tausif Diwan, "Survey on Tweet Segmentation and Sentiment Analysis", *International Journal of Computer Sciences and Engineering*, Online ISSN: 2347-2693, pp. 391-394, Vol. 6, No. 1, 31st January 2018, Published by ISROSET, DOI: 10.26438/ijcse/v6i1.391394, Available: https://www.ijcseonline.org/full_paper_view.php?paper_id=1690.