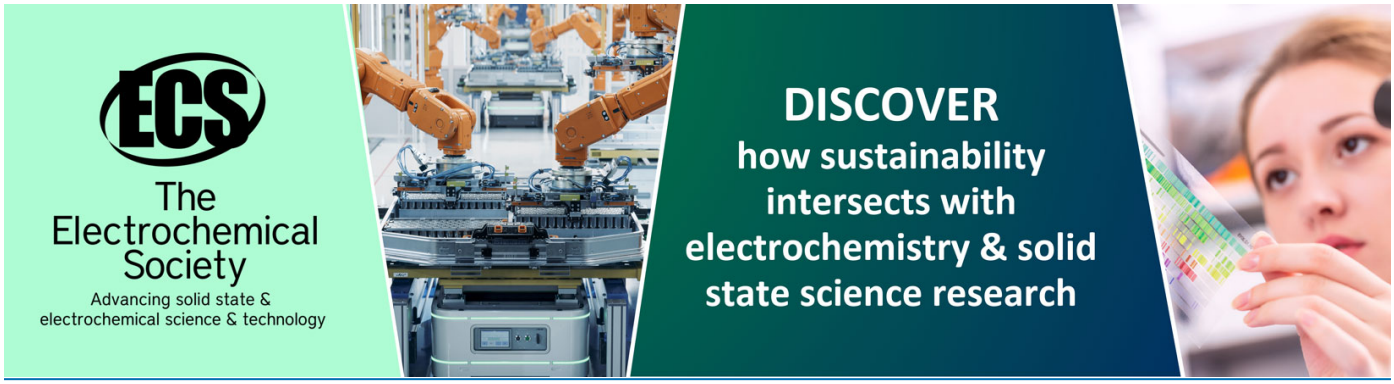
Knowledge-oriented Hierarchical Neural Network for Sentiment Classification



Knowledge-oriented Hierarchical Neural Network for Sentiment Classification

Yanliu Wang[1] and Pengfei Li[2, *]

[1] School of Smart City, Beijing Union University, No.97 Beisihuan East Road, Chao Yang District, Beijing 100101, China
[2] School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, 639798, Singapore

[*]E-mail: pli006@e.ntu.edu.sg

Abstract. Sentiment classification aims to classify the sentimental polarities of given texts. Lexicon-based approaches utilize lexical resources to explore the opinions according to some specific rules, whose effectiveness strongly depends on the goodness of the lexical resources and the rules. Traditional machine-learning methods tightly rely on feature engineering and external NLP toolkits with unavoidable errors. Deep learning models strongly rely on a large amount of labelled data to train their numerous parameters, which often suffer from overfitting issue since it is difficult to obtain sufficient training data. To address the issues, we design a model that combines Knowledge-oriented Convolutional Neural Network (K-CNN) and bidirectional Gated Recurrent Neural Network (biGRU) in a hierarchical way for sentiment classification. Firstly K-CNN is used to capture the n-gram features in sentences. Sentiment word filters are constructed in the knowledge-oriented channel of K-CNN based on the linguistic knowledge from SentiWv ordNet, which can capture the sentiment lexicons and alleviate overfitting effectively. Then biGRU with attention mechanism is utilized to model the sequential relations between sentences and obtain the document-level representation based on the relevance of each sentence to the final sentiment classification. Experiments on two datasets show that our model outperforms other classical deep neural network models.

1. Introduction

The overwhelming of text data on the Internet leads to the revolution of automatically exploiting and extracting valuable information from them. Sentiment classification is a subfield of natural language processing (NLP) which is designed for automatically determining the overall sentiment orientation of the given document based on the opinions in it. The task is to classify the given document into different sentimental polarities (e.g., positive and negative), depending on the contents. With the rapid growth of available subjective texts on the Internet in the form of product reviews, blog posts and comments in discussion forums, business analysts are turning their eyes on the Internet in order to obtain factual as well as more subtle and subjective information (opinions) on companies and products [1]. Therefore, sentiment classification is becoming progressively crucial along with the data blooming.

Current approaches for sentiment classification include lexicon-based and machine-learning-based methods. Lexicon-based techniques generally use the opinion words in the dictionary combined with complex rules and work on the assumption that the collective polarity of a sentence or a document is

the sum of polarities of the individual phrases or words [2]. However, with the difficulties of constructing and maintaining the expert rules, the precision and recall obtained by lexicon-based models are not satisfied. In machine-learning-based approaches, sentiment polarities are automatically inferred from a large amount of labelled data. In early years (before 2013), people construct lexical, syntactic and semantic features by complex feature engineering, where the handcrafted features are specifically designed and adjusted for the specific domain. The performance is limited by extensive feature engineering, domain dependency, and unavoidable errors of external NLP toolkits. Recently, researchers started to focus on developing deep compositional models such as Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) for sentiment classification [3-6]. Deep learning models have demonstrated excellent results because the features are automatically extracted by learning from a large amount of training data, and they are more adept at understanding complicated semantics of a document. However, deep learning models typically have a large number of free parameters which are required to train with a lot of labelled data. It is tough to obtain sufficient training data that covers all the variants of sentiment expressions due to the complexity of natural language. With the limited training data, deep learning models frequently suffer from overfitting issue which hinders their performances.

To tackle the problems mentioned above, we take the advantages of both lexicon-based and machine-learning-based methods and propose a Knowledge-oriented Hierarchical Neural Network (KHNN) that combines Knowledge-oriented CNN (K-CNN) [7] and bidirectional GRU (biGRU) [8] hierarchically for sentiment classification. The idea of hierarchical modelling of the document comes from the theory that the meaning of a long expression is determined by its compositions [9]. KHNN first extracts n-gram features in sentence-level using K-CNN, which reflect the sentimental polarities of sentences; then a bidirectional GRU is used to model the sequential and semantic relations between sentences in the document; finally, a document-level representation is obtained by combining the biGRU outputs using attention mechanism based on the relevance of each sentence to the final sentiment classification. The contributions of the paper are summarized as follows:

- Sentiment word filters in K-CNN are built based on SentiWordNet [10], which can adequately capture the words with strong sentimental polarities appeared in the sentences. Meanwhile, with the fixed weights of sentiment word filters, the number of free parameters is significantly reduced, which effectively minimizes the reliance on training data to alleviate overfitting.
- The proposed model is constructed hierarchically by combining K-CNN which focuses on local features, and biGRU with attention mechanism which captures the global features by sequentially combing the local features. To be specific, K-CNN with sentiment word filters is designed to capture the linguistic clues about the sentiment in each sentence. Then a biGRU with attention mechanism is used to capture the sequential relation and semantic relatedness between sentences to obtain the overall sentiment representation of the document.
- Real-world datasets including IMDB movie reviews and Amazon apparel reviews are used to evaluate our model. Results show that our model can adequately capture the sentimental polarities of documents and outperforms other classical deep learning models for sentiment classification.

2. Related work

The approaches for sentiment analysis are divided into two categories: lexicon-based and machine-learning-based methods. The lexicon-based approaches use the words in the dictionaries and base on some complicated rules. The machine-learning methods include traditional machine-learning and deep learning methods. The difference is that complicated feature engineering is necessary for traditional machine-learning methods. However, it is automatically fulfilled in deep learning methods.

For the lexicon-based method, some works presented approaches which detect the sentiment polarity of a text depending on the semantic orientation of words or phrases appearing in the document [11, 12]. Also, Hanen and Salma [13] presented a lexicon-based approach using lexicon words to determine the sentimental orientation of Facebook comments. The accuracy of the lexicon-based method is not satisfied because it is tough to construct and maintain complicated rules. The machine-

learning-based approach performs better since sentiment polarities can be automatically inferred from a large amount of labelled data. Some early works focused on creating effective features and used Naive Bayes and support vector machine (SVM) for sentiment classification [14, 15, 16]. However, sophisticated feature engineering is indispensable, and the performance of such system strongly depends on the quality of designed features. Recently, deep learning is becoming more and more popular because the features are automatically extracted from input text by learning from a large amount of data. Convolution Neural Network (CNN) and Recurrent Neural Network (RNN) are footstones for the sentiment classification tasks. Dos Santos et al. [17] designed a model, which combines two CNN layers and Wang et al. aggregate CNN and RNN for sentiment classification [18]. Some works focused on hierarchical neural network models for sentiment classification [19, 20], which perform state-of-the-art accuracy. Even though the deep learning models can achieve high performance in sentiment classification, they rely on quite a large amount of labelled data, which are hard to obtain. Hence, the overfitting problem often perplex researchers.

In order to addressee the issues of lexicon and machine-learning-based approaches, some researchers took a benefit of both methods and proposed some hybrid approaches for sentiment analysis. Zhang et al. [21] firstly use the lexicon-based method for entity-level analysis and then they use additional opinionated indicators to analyze the result of the previous lexicon-based method. Shin et al. [22] suggested a sentiment analysis method that integrates lexicon embeddings and an attention mechanism into Convolutional Neural Networks. In addition, similar hybrid models were proposed and achieved high accuracy for sentiment classification [23, 24]. Therefore, our concentration is on the hybrid method with a novel hierarchical model structure.

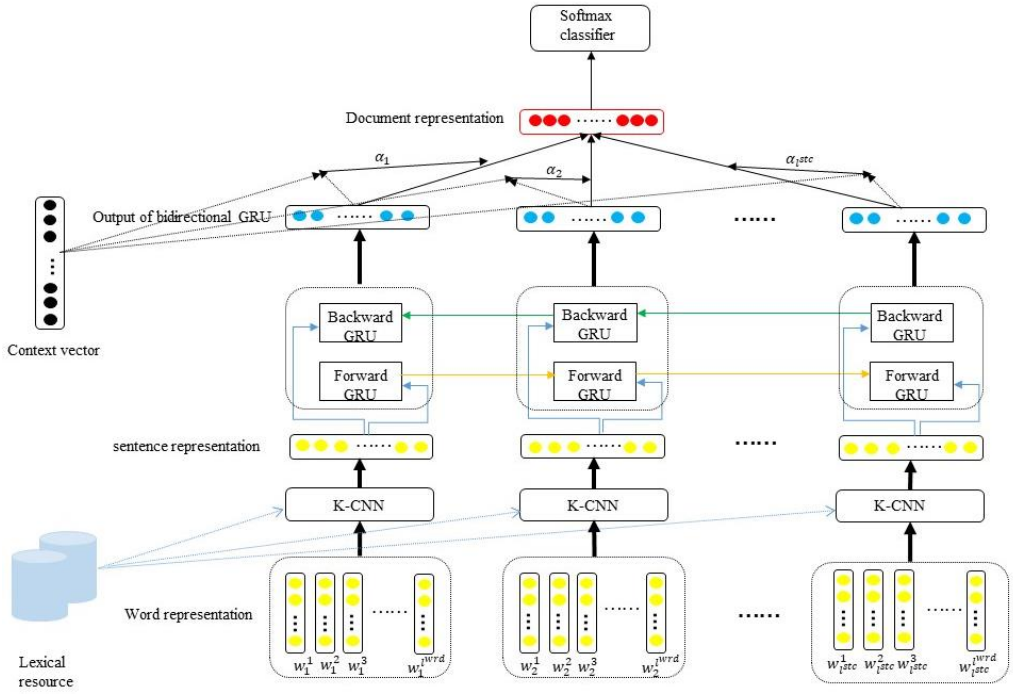


figure 1. The overall structure of the Knowledge-oriented Hierarchical Neural Network (KHNN). w_i^j is the vector representation of the j-th word in the i-th sentence, l^{stc} is the number of sentences in the document, l^{wrd} is the number of words in each sentence and α_i is the weight for the i-th sentence representation vector in attention mechanism.

3. Methodology

The proposed Knowledge-oriented Hierarchical Neural Network (KHNN) has two main components, which are the Knowledge-oriented Convolution Neural network (K-CNN) and bidirectional Gated Recurrent Neural Network (biGRU) with an attention mechanism. K-CNN extracts local features such

as significant sentiment lexicons or linguistic clues about sentimental polarity from sentences and obtains the sentence representations. biGRU captures the sequential semantic relations between sentences and obtains the final document representation through an attention mechanism. Finally, the model detects the sentimental polarity based on global features. The overall model architecture is shown in Figure 1.

3.1. Local feature extraction using K-CNN

The main difference between K-CNN and conventional CNN is that K-CNN contains additional knowledge-oriented channel, whose convolutional filters (word filters) are constructed from prior human knowledge instead of learning from data. The benefit of K-CNN is that linguistic knowledge is incorporated into the model which allows it to capture the linguistic clues of sentimental polarity more effectively. Also, K-CNN alleviates overfitting issue since the number of free parameters is significantly reduced. The detailed structure of K-CNN is shown in Figure 2.

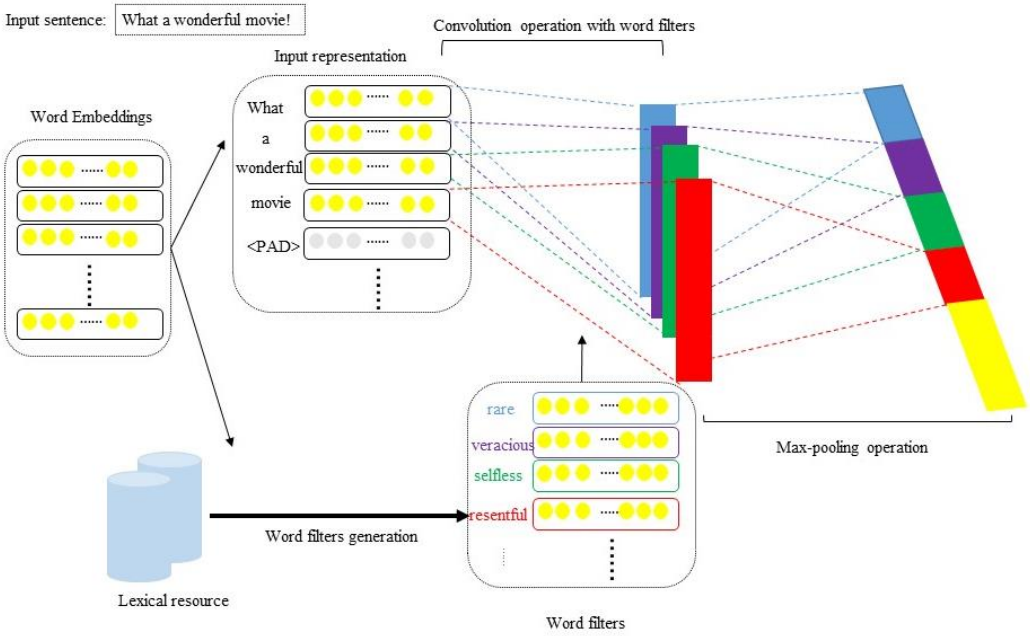


figure 2. The structure of knowledge-oriented channel of K-CNN

3.1.1. Sentence representation. The input document D is composed of a sequence of sentences $D = \{s_1, s_2, s_3, \dots\}$, and each sentence consists of a sequence of words. We first tokenize each sentence s_i into word tokens with its lower case $s_i = \{x_1, x_2, x_3, \dots\}$. Next, we map each word into a continues vector space using pre-trained word embeddings to capture the semantic and syntactic information of words. To be more specific, each word x_i is represented as a word vector $w_i \in R^d$ according to word embedding matrix $W^{word} \in R^{d*|s|}$, where $|s|$ is the size of the vocabulary and d is the dimensionality of word embeddings. Since our model only receives the fixed length vectors as input, the lengths of words in each sentence and sentences in the document are padded as l^{wrd} and l^{stc} using special padding character <PAD> with zero word embedding vector, where l^{wrd} is the maximum number of words in a sentence and l^{stc} is the maximum number of sentences in a document. Finally, a document is represented as $D = \{e_1, e_2, \dots, e_{l^{stc}}\}$, where e_i is the vector representation of sentence s_i: $e_i = \{w_1, w_2, \cdots, w_{lwrd}\}$.

3.1.2. Sentiment word filters generation. SentiWordNet [10] is one of the most frequently utilized and advantageous lexical resources for sentiment analysis. It is a publicly available dictionary based on WordNet [25] which assigns to each WordNet synset three sentiment scores (positivity, negativity,

objectivity) ranging from 0 to 1. SentiWordNet includes a relatively large number of terms compared with other popular lexicons [26]. In addition, SentiWordNet provides polarity scores to each sense of a term, whereas the other lexicons only cover a single polarity score. Based on SentiWordNet, we extract significant sentiment lexicon whose sentimental score (either positivity or negativity) is greater than or equal to 0.8. Table 1 shows examples of positive and negative lexicons extracted from SentiWordNet.

Table 1. Examples of lexicons extracted from SentiWordNet

positive	selfless attractive pretty great gracious admirable fabulous honorable superb excellent reputable intellectual good veracious
negative	wrong malevolent unreliable troublesome low depressed foul incompatible unfortunate improper negative miserable fouled

After the extraction of sentiment lexicons, each sentiment lexicon is transformed into a sentiment word filter $\boldsymbol{f} \in \boldsymbol{R^d}$ by looking up the same embedding matrix $\boldsymbol{W^{word}}$ as used for input words. The sentiment word filters are used as the convolutional filters in the knowledge-oriented channel of K-CNN to capture the significant sentiment lexicons appeared in the sentence.

3.1.3. Convolution operation. Given the input sentence matrix is $e_i = \{w_1, w_2, \cdots, w_{lwrd}\} \in R^{d*l^{wrd}}$, convolution operation is performed on the input sentence using convolutional filters $f_i \in R^{d*k}$, where k is the convolution window size which also corresponds to the k-grams in input texts. In the knowledge-oriented channel, the convolutional filters f_i are the sentiment word filters whose weights are pre-defined and $k = 1$ to capture the words with high sentimental polarities. In the data-oriented channel, the weights of f_i are random initialized and adjusted by learning from data through back-propagation. We set convolution window $k = 3$ to capture the long-term dependencies between words and other sentiment lexicons which could be ignored in the knowledge-oriented channel.

Mathematically, a feature map $\boldsymbol{m} \in \{\boldsymbol{m_1}, \boldsymbol{m_2}, \cdots, \boldsymbol{m_{l^{wrd}-k+1}}\}$ is obtained for each convolutional filter $\boldsymbol{f}$, where $\boldsymbol{m_i}$ is calculated from input words of window size $\boldsymbol{k}$ $\boldsymbol{e_i}[*, \boldsymbol{i}: \boldsymbol{i}+\boldsymbol{k}]$ following equation (1).

$$m_i = \sigma(\sum(e_i[*, i: i+k] \odot f) + b) \quad (1)$$

where $\boldsymbol{\sigma}$ is a non-linear function, such as tanh and relu, $\boldsymbol{b}$ is the bias and $\odot$ is the Hadamard product between two matrices.

After the convolution operation, max-pooling is performed on each feature map to capture the most significant feature. We use global max pooling as shown in equation (2).

$$p = \max\{m\} = \max\{m_1, m_2, \cdots, m_{l^{wrd}-k+1}\} \quad (2)$$

With the complementary effects of the knowledge-oriented channel and data-oriented channel, K-CNN has outstanding performance in extracting local n-gram features from each sentence in the document.

3.2. Global feature extraction using biGRU

Apart from the sentiment feature vector of each sentence which represents the sentiment information of a sentence obtained via K-CNN, the sequential information between a sequence of sentiment feature vectors also affects the sentimental polarity of a document. For example, “I hated fiction movies because they are ridiculous and boring. However, one day I reverse my opinion after watching Avengers: Infinity War”. If we do not consider such sequential information between sentences, the result of final sentiment classification may be wrong. Hence biGRU with attention mechanism is utilized to capture the sequential information and obtain the global feature vector in document-level.

3.2.1. Bi-directional GRU to model document. GRU [27] is a type of RNN which is designed to model sequential data, where the hidden state h_t at time step t depends on the current input x_t as well as a hidden state at time step h_{t-1}. To overcome the gradient vanishing and exploding [28] problem of

RNN, two special gates including update gate and reset gate are introduced. The detailed process of GRU is shown as following:

$$r_t = \sigma(V_r * x_t + W_r * h_{t-1} + b_r) \quad (3)$$

$$z_t = \sigma(V_z * x_t + W_z * h_{t-1} + b_z) \quad (4)$$

$$\widehat{h_t} = tanh(V * x_t + r_t(W \odot h_{t-1}) + b_h) \quad (5)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widehat{h_t} \quad (6)$$

where $\boldsymbol{\sigma}$ is a sigmoid function, $\odot$ represents the element-wise multiplication, $\boldsymbol{r_t}$ represents the reset gate, $\boldsymbol{z_t}$ represents the update gate and $\widehat{\boldsymbol{h_t}}$ represents the candidate hidden layer.

In our model, we use a bi-directional GRU [8] which stacks two GRUs: forward GRU and backword GRU. The main idea is that the output of each time step may not only depend on the previous elements of the sequence but also depends on the subsequent elements. The forward GRU processes the input sentence from $\boldsymbol{e_1}$ to $\boldsymbol{e_{l^{stc}}}$ obtaining the forward hidden states $\overrightarrow{\boldsymbol{h_t}} = \overrightarrow{\boldsymbol{GRU}(\boldsymbol{e_t})}$ where $\boldsymbol{t} \in [\boldsymbol{1}, \boldsymbol{l^{stc}}]$. The backward GRU processes the input from $\boldsymbol{e_{l^{stc}}}$ to $\boldsymbol{e_1}$ calculating the backward hidden states $\overleftarrow{\boldsymbol{h_t}} = \overleftarrow{\boldsymbol{GRU}(\boldsymbol{e_t})}$ where $\boldsymbol{t} \in [\boldsymbol{l^{stc}}, \boldsymbol{1}]$. Then, we concatenate the hidden states at each time step to obtain the sentence representation $\boldsymbol{h_t} = [\overrightarrow{\boldsymbol{h_t}}, \overleftarrow{\boldsymbol{h_t}}]$.

3.2.2. Combine biGRU outputs using the attention mechanism. Yang et al. [29] proposed a model with an attention mechanism for constructing the document representation according to the semantic importance of contents. The motivation is that not all the sentences play the same role to the overall sentiment polarity and the same sentence may play a different role in a distinct context. With the attention mechanism, we combine the bi-direction GRU outputs h_i at each time step to obtain a global document-level feature vector. The calculation process is as follows:

$$u_i = tanh(W_s * h_i + b_s) \quad (7)$$

$$\alpha_i = \frac{e^{u_i{}^T * u_s}}{\sum_i e^{u_i{}^T * u_s}} \quad (8)$$

$$d = \sum_i \alpha_i * h_i \quad (9)$$

$\boldsymbol{u_i} \in \boldsymbol{R^{\alpha}}$ is the hidden representation of $\boldsymbol{h_i}$ by passing $\boldsymbol{h_i}$ to a fully connected neural network. $\boldsymbol{u_s} \in \boldsymbol{R^{\alpha}}$ is a context vector which is used to calculate the importance of each sentence ($\boldsymbol{\alpha_i}$) for the overall sentiment orientation, and $\boldsymbol{d}$ is the final document representation. The context vector is randomly initialized and learned during the training process.

3.3. Regularization and sentiment classification

Firstly, we apply dropout regularization [30] to our model for alleviating it from overfitting. The main idea is to randomly drop the connection between units from the neural network during training. After the dropout operation, the obtained final feature vector is fed into a softmax classifier to get the probability distribution over sentiment categories $\boldsymbol{p(D)}$.

We use categorical cross-entropy loss function to calculate the loss between real sentiment distribution $\boldsymbol{p^r(D)}$ and output distribution $\boldsymbol{p(D)}$, as shown in equation (12):

$$\text{loss} = -\sum_{D \in T} \sum_{i=1}^{c} p_i^r(D) \log p_i(D) \quad (10)$$

where T is the training dataset, $\boldsymbol{c}$ is the number of sentiment classes. $\boldsymbol{p^r(D)}$ is a $\boldsymbol{c}$ dimensional one-hot vector whose element corresponding to the real document sentiment category is 1 and other elements are set as 0. Adam [31] optimizer is used to minimize the loss function.

4. Experiment

4.1. Dataset description

We use two real-world sentiment classification datasets to evaluate our model. One dataset IMDB[3] which contains 50,000 movie reviews with high sentimental polarities [32]. Training set and test set are separated equally with balanced positive and negative movie reviews. Another dataset is apparel reviews from Amazon[4] which consists of 1000 positive and 1000 negative reviews [33]. For IMDB dataset, the average number of words in a sentence and number of sentences in a review is 90 and 14 respectively. Whereas for Amazon apparel review dataset, the average number of words in a sentence and number of sentences in a review is 60 and 5 respectively. Hence, Amazon review dataset is relatively smaller compared with IMDB dataset.

4.2. Experiment settings

To investigate the effectiveness of our proposed KHNN model, we compare KHNN with several baseline models including non-hierarchical models and hierarchical models. For non-hierarchical models, we compare the performances of CNN, LSTM, GRU, and bi-directional GRU (biGRU) with an attention mechanism. For hierarchical models, we study biGRU(atten)+CNN which is biGRU with attention for sentence modeling and CNN for document modeling, CNN+biGRU(atten) which is CNN for sentence modeling and biGRU with attention for document modeling, as well as our proposed KHNN which is K-CNN for sentence modeling and biGRU with attention for document modeling.

For CNN, the number of convolutional filters is set to 100, and the window size of the filter is 3. For LSTM, GRU, and biGRU, the dimension of hidden units is 128. In the data-oriented channel of K-CNN, the number of convolutional filters is set to 50 for IMDB and 75 for Amazon reviews since less trainable filters are needed for K-CNN. In the attention mechanism, the dimension of the context vector is set to 100 for IMDB dataset and 50 for Amazon review dataset. Batch size is 100 for IMDB and 50 for Amazon reviews. Word2Vec [34] word embeddings pre-trained on Google News dataset with about 100 billion words are used for all the models. The dropout rate is set to 0.3, and we train each model for 15 epochs.

The evaluation metrics we used include accuracy, macro-average precision, recall, and F1 score. For IMDB, we evaluated our model on test set; for Amazon reviews dataset, we used 5-fold cross-validation to evaluate the models since no train-test split is provided. We report the average results of the 5 runs.

4.3. Experiment results and analyses

The experiment results of all the models described in Section 4.2 on IMDB dataset and apparel reviews dataset are shown in Table 2.

Table 2. Results of various models for sentiment classification on IMDB and Amazon review dataset. "atten" means attention mechanism which is associated with biGRU.

	IMDB				**Amazon**			
Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)
CNN	88.02	88.38	88.02	87.99	74.60	75.05	74.60	74.49
LSTM	85.45	85.56	85.45	85.44	55.25	55.67	55.25	54.32
GRU	87.65	87.75	87.65	87.64	57.20	57.62	57.20	56.62
biGRU(atten)	89.10	89.46	89.10	89.07	68.85	69.45	68.85	68.63
biGRU(atten)+ CNN	90.08	90.18	90.08	90.08	84.95	85.16	84.95	84.93
CNN+biGRU(atten)	90.23	90.24	90.23	90.23	87.90	87.97	87.90	87.90
Our KHNN	**90.90**	**90.92**	**90.90**	**90.90**	**88.75**	**88.88**	**88.75**	**88.74**

[3] The dataset can be downloaded from http://ai.stanford.edu/ amaas/data/sentiment/

[4] The dataset can be downloaded from http://www.cs.jhu.edu/ mdredze/datasets/sentiment/

4.3.1. Comparison of non-hierarchical models. We first study the performances of non-hierarchical models according to the experiment results shown in Table 2. CNN has demonstrated better performance than LSTM and GRU. One reason is that CNN is good at capturing n-gram features such as words and phrases expressing high sentimental polarities, which are particularity important for sentiment classification. Another reason is due to the limitation of LSTM and GRU in dealing with long sequences: even through gate operations are introduced to alleviate gradient vanishing problem, the models still cannot cope with long sequence modelling, resulting in information loss for document-level sentiment classification. However, GRU outperforms LSTM due to the fewer model parameters, which is advantageous in training with small datasets.

To address the limitation of GRU, bidirectional modelling and attention mechanism are added to GRU, and significant improvement is observed. biGRU considers not only forward sequences but also backward sequences, which considers the whole context information. Attention mechanism combines biGRU outputs based on the contribution of each hidden state to the final sentiment classification. This solves the gradient vanishing problem since every time step is considered for the final document representation. Therefore, in hierarchical models, we only consider biGRU with attention mechanism for RNN-based models.

4.3.2. Comparison of hierarchical models. Based on the results in Table 2, hierarchical models yield better performance than the non-hierarchical models, which proves that hierarchical structure is indeed beneficial in dealing with document-level sentiment analysis since hierarchical models process texts from sentence-level to document-level and preserve their semantic compositionality. Results show that CNN+biGRU(atten) performs better than biGRU(atten)+CNN. The reasons are that CNN is good at capturing local n-gram features instead of global features, and biGRU with attention mechanism does well in extracting global features but may not be able to capture n-gram features that are important for sentiment classification effectively. Hence, it is beneficial to firstly using CNN to capture local n-gram features in each sentence and then utilize biGRU with an attention mechanism to extract a global feature from the sequential sentence representations.

Comparing our KHNN model with CNN+biGRU(atten), the performance improvement shows that K-CNN with sentiment word filters is more effective than conventional CNN in capturing local features from sentences. Since the sentiment word filters are generated from SentiWordNet lexicons with high sentimental polarities, they can adequately capture such lexicons appeared in the sentence and alleviate overfitting issue.

5. Conclusion

In this paper, we propose a Knowledge-oriented Hierarchical Neural Network (KHNN) for sentiment classification, where a document is modelled from sentence-level to document-level. A Knowledge-oriented CNN (K-CNN) is used to capture n-gram local features from sentences. Sentiment word filters are constructed in the knowledge-oriented channel based on SentiWordNet, which can effectively capture lexicons with high sentimental polarities and alleviate overfitting issue. A bidirectional GRU with attention mechanism is used to model the sequential semantics between sentences and obtain a document-level global representation for sentiment classification. Experiments on two real-world datasets show that our model yields better results than other classical deep neural networks.

In future work, more external knowledge will be explored and incorporated into KHNN to model the document. Furthermore, we plan to apply KHNN to more challenging tasks such as aspect-based sentiment analysis.

References

[1] Alaa Hamouda, Mohamed Rohaim, 2011. Reviews Classification Using SentiWordNet Lexicon. *The Online Journal On Computer Science and Information Technology (OJCSIT)* **Vol. (2)** - No. (1).

[2] Devika M D, Sunitha C, Amal Ganesh, 2016. Sentiment Analysis: A Comparative Study on Different Approaches. In: *Fourth International Conference on Recent Trends in Computer Science & Engineering*. *Procedia Computer Science* 87 (2016) 44 – 49.
[3] Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu, 2016. Neural sentiment classification with user and product attention. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*.
[4] Xu J, Chen D, Qiu X, and Huang X, 2016. Cached long short-term memory neural networks for document-level sentiment classification. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2016).*
[5] Monalisa Ghosh and Goutam Sanyal, 2018. Document Modeling with Hierarchical Deep Learning Approach for Sentiment Classification. In: *Proceedings of the 2nd International Conference on Digital Signal Processing*. Pages 181-185
[6] Guozheng Rao, Weihang Huang, Zhiyong Feng, and Qiong Cong, 2018. LSTM with sentence representations for document-level sentiment classification. In: *Neurocomputing*, **Volume 308**, Pages 49-57.
[7] Pengfei Li, Kezhi Mao, 2018. Knowledge-oriented convolutional neural network for causal relation extraction from natural language texts. *Expert Systems with Applications.* **Volume 115** (2019).Pages 512–523.
[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473.*
[9] Gottlob Frege. 1892. *On sense and reference*. Ludlow (1997), pages 563–584.
[10] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani, 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation, LREC '10*, Valletta, Malta, pages 2200–2204.
[11] Peter D. Turney, 2002. Thumbs up or Thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: *Proceedings of 40th Annual Meeting of the Association of Computational Linguistics*, Philadelphia, PA, pages 417–424.
[12] Maite Taboada, Caroline Anthony, and Kimberly Voll, 2006. Method for Creating semantic orientation dictionaries. In: *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC)*, Genoa, Italy, pages 427–432.
[13] Hanen Ameur, Salma Jamoussi, 2013. Dynamic construction of dictionaries for sentiment classification. In: *13th IEEE International Conference on Data Mining Workshops. ICDM workshops,* TX, USA (2013), pages 896-903.
[14] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan, 2002. Thumbs up? sentiment classification using machine learning techniques. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Philadelphia, July 2002, pages. 79-86.
[15] Bingwei Liu, E. Blasch, Yu Chen, Dan Shen, and Genshe Chen, 2013. Scalable sentiment classification for Big Data analysis using Naïve Bayes Classifier. In: *Big Data, 2013 IEEE International Conference,* Silicon Valley, CA, 2013, pages. 99-104.
[16] Joseph D. Prusa, Taghi M. Khoshgoftaar, and David J. Dittman, 2015. Impact of Feature Selection Techniques for Tweet Sentiment Classification. In: *Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference*, pages 299-304.
[17] Cicero Nogueira dos Santos and Maira Gatti, 2014. Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts. In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics*: Technical Papers, Dublin, Ireland, pages 69–78.
[18] Xingyou Wang, Weijie Jiang, and Zhiyong Luo, 2016. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*: Technical Papers, Osaka, Japan, pages 2428–2437.

[19] Duyu Tang, Bing Qin, and Ting Liu, 2015. Document modeling with gated recurrent neural network for sentiment classification. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* Lisbon, Portugal, pages 1422-1432.

[20] Sebastian Ruder, Parsa Ghaffari, and John G. Breslin, 2016. A Hierarchical Model of Reviews for Aspect-based Sentiment Analysis. *arXiv preprint arXiv:1609.02745.*

[21] Lei Zhang, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, and Bing Liu, 2011. Combining Lexicon-based and Learning-based Methods for Twitter Sentiment Analysis. *Technical Report HPL*-2011-89, HP, 21/06/2011.

[22] Bonggun Shin, Timothy Lee, and Jinho D. Choi. 2016. Lexicon Integrated CNN Models with Attention for Sentiment Analysis. *arXiv preprint arXiv:1610.06272.*

[23] Vikash Nandi, Suyash Agrawal. 2016. Political Sentiment Analysis using Hybrid Approach. In: *International Research Journal of Engineering and Technology*, **Vol.3**, issue. 5, pages 1621-1627.

[24] Amira Shoukry, Ahmed Rafea. A Hybrid Approach for Sentiment Classification of Egyptian Dialect Tweets *2015. In: First International Conference on Arabic Computational Linguistics (ACLing)*, Cairo, Egypt, pages 78-85.

[25] G. A. Miller, 1995. Wordnet: A lexical database for English. In: *Communications of the ACM*, **(11)**:39–41. http://wordnet.princeton.edu/.

[26] Liviu Adrian Cotfas, Camelia Delcea, Irina Raicu, Ioana Alexandra Bradea, and Emil Scarlat, 2017. Grey sentiment analysis using SentiWordNet. In: *2017 IEEE International Conference on Grey Systems and Intelligent Services, GSIS 2017*, pages 284-288.

[27] Kyunghyun Cho Bart van Merrienboer Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares Holger Schwenk, and Yoshua Bengio, 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724-1734.

[28] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning Long-Term Dependencies with Gradient Descent is Difficult. In: *IEEE Transaction on Neural Networks*, **Vol**. 5, NO.2., pages 157-166.

[29] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy, 2016. Hierarchical Attention Networks for Document Classification. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*: Human Language Technologies, pages 1480–1489.

[30] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In: T*he Journal of Machine Learning Research 15 (2014)*, pages 1929-1958

[31] Diederik P. Kingma, Jimmy Ba, 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*

[32] Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C, 2011. Learning word vectors for sentiment analysis. In: *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 142–150.

[33] Blitzer, J.; Dredze, M.; and Pereira, F, 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain adaptation for sentiment classification. In: *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics. June 23-30, 2007, Prague, Czech Republic.*

[34] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, 2013. Efficient estimation of word representations in vector space. In: P*roceedings of International Conference on Learning Representations (ICLR)*, pages 1-12.