# A Survey on Modularization of Chatbot Conversational Systems

Xinzhi Zhang, Shiyulong He, Zhenfei Huang, and Ao Zhang(✉)

College of Intelligence and Computing, Tianjin University, Tian Jin, China
azhang@tju.edu.cn

**Abstract.** Chatbots have attracted more and more attention and become one of the hottest technology topics. Deep learning has shown excellent performance in various fields such as image, speech, natural language processing and dialogue, it has greatly promoted the progress of chatbots, it can use large amounts of data to learn response generation and feature representations. Due to the rapid development of deep learning, hand-written rules and templates were quickly replaced by end-to-end neural networks. Neural networks is a powerful model that can solve generation problems in conversation response. People's requirements for chatbots have also increased with the continuous improvement of neural network models. In this article, we discuss three main technologies in chatbots to meet people's requirements, syntax analysis, text matching and sentiment analysis, and outline the latest progress and main models of three technologies in the field of chatbots in recent years.

**Keywords:** Chatbots · Neural networks · Deep learning

## 1 Introduction

Nowadays, with the rapid development of intelligent question and answer system, chatbot has been gradually integrated into all aspects of human society. Whether in academia, industry or People's Daily lives, we can always see a variety of chatbots make outstanding contributions. In recent years, the continuous innovation and expansion of artificial intelligence, neural network and deep learning have greatly improved the functions and features of open domain chatbots. As a result, humans have the ability to communicate more smoothly with machines, and chatbots behave more like humans.

As a system for communicating directly with humans, we are also increasingly demanding of chatbots. For task-oriented robots that help us achieve a certain goal (most of which exist now), we require the accuracy and validity of the information; In contrast, most of the non-task robots used for small talk require semantic correctness while also paying attention to the rationality and consistency of response when communicating with people. In recent years, great progress has been made in the research and technology of search-based and generative chatbots, which also provides us with broader ideas for relevant exploration.


Y. Nah et al. (Eds.): DASFAA 2020 Workshops, LNCS 12115, pp. 175–189, 2020.
https://doi.org/10.1007/978-3-030-59413-8_15

For these models, the existing research mainly focuses on the overall technology, but the overall technology can be further divided into multiple modules, among which the most important module technologies can be divided into three: syntactic analysis, text matching and emotion analysis. Syntactic analysis is the basic work of natural language processing. It analyzes the syntactic structure of a sentence (subject-verb-object structure) and the dependencies between words (juxtaposition, subordination, etc.). Syntactic analysis can lay a solid foundation for the application scenarios of natural language processing, such as semantic analysis, affective tendency and thought extraction. Through emotion analysis, the advantages and disadvantages of products in various dimensions can be explored to determine how to improve products. Emotion analysis based on deep learning has the advantages of high accuracy, strong universality and no need of emotion dictionary. Text matching is mainly divided into "deep learning model based on single-turn response matching" and "deep learning model based on multi- turn response matching". Based on the deep learning model of single-turn response matching, the two documents to be matched are mapped to two vectors, and then the two vectors are transferred through the neural network to output the results, and the conclusion of whether they match is drawn. The feature of the deep learning model based on multi- turn response matching is that the two documents to be matched are expressed as words, phrases, sentences and other different granularity through the neural network, and then the similarity matrix is crossed and input into the neural network to obtain the conclusion whether they match or not.

The important modules are always essential, they will not always have the overall architecture of the big change, in this case, through a theoretical analysis of the techniques used in the three modules, the understanding of performance, accuracy, effect and analysis, refining the overall performance of the master in pairs, deep understanding of the structure of the various modules and the reasons of the different performance, which can be more effective from partial optimization to enhance and improve the overall performance.

In this paper, we will review and summarize relevant researches of the above three modules in recent years. In the following sections, we describe the characteristics and applications of these three modules in more detail.

## 2 Syntactic Analysis

Syntactic analysis is one of the core problems in language comprehension and has received extensive attention. Dependency parsing is a popular method to solve this problem, because there are dependency tree libraries in many languages (Buchholz et al. 2006; Nivre et al. 2007; McDonald et al. 2013) [1] and dependency parsers.

Chen and Manning et al. (2014) [2] proposed a neural network version of the parser based on greedy transformation. In their model, feedforward neural networks with a hidden layer are used to make transition decisions. The hidden layer has the ability to learn any combination of atomic inputs, eliminating

the need for manual design features. In addition, because the neural network adopts distributed representation, the similarity between lexical markers and arc markers can be modeled in continuous space. Despite their model than the greedy manual design of similar products better, but it cannot compete with the most advanced dependency parser, which is trained for a structured search. Greed model although extremely fast, but usually run into search errors, because they are unable to recover from the wrong decision.

David Weiss and Chris Alberti et al. (2015) [3] proposed a structured perceptron training method based on dependency analysis based on neural network transformation, combined the representation ability of neural network with structured training and advanced search of reasoning support, and used a large number of sentences expanded by automatic analysis corpus to learn the neural network representation. This work started from the basic structure of Chen and Manning et al. (2014), but there was a further improvement in the architecture and optimization process: allowing smaller POS tags to be embedded and Relu units to be used in the hidden layer. These improvements improved the accuracy of the model by nearly 1% compared with Chen and Manning et al. (2014) (Fig. 1).

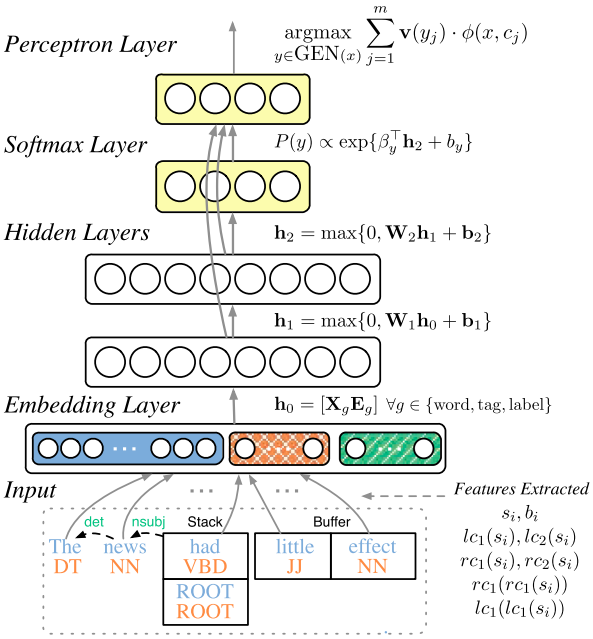


**Fig. 1.** Representation model framework.

In addition, they also proposed an effective method to use unmarked data, called "three-training" method. They used two different parsers to parse the

unmarked data, and only selected two parsers to generate statements of the same tree, thus generating a large number of trusted parse trees.

In the same year, Mingbo Ma, Liang Huang et al. (2015) [4] proposed a very simple convolutional neural network (DCNNs) based on dependence, which is similar to the sequential CNNs model of Kim et al. (2014) [5], but the difference is that Kim et al. (2014) put a word in the sequence of the text, the model will be a word and its father, grandfather and great grandfather, and brothers and sisters in dependence on the tree, in this way to integrate information over a long distance. This is a very simple correlational convolution framework that performs better than sequential CNNs at sentence modeling.

Tao, Ji, Yuanbin Wu et al. (2019) [6] proposed a map neural network (GNNs) to learn to represent dependency tree nodes, said will map neural network is added into dependency parsing, efficient higher order information coding to rely on the representation of a tree node in: given a sentence, a parser for all the words to score in the first place, see if they can keep effective dependencies, and then use the decoder (for example, greed, maximum spanning tree) generated from these marks a complete parsing tree. Two previous outstanding works on node performance were recursive neural networks (RNNs) (Kiperwasser and Goldberg et al. (2016)) and biaffine mappings (Dozat and Manning et al. (2017)), but these representations ignore the characteristics associated with dependency structures.

Given a weighted graph, a GNN is embedded in a node by recursively aggregating the nodes of its neighbors, and the graph can be modified during parsing. The representation of a node through the superposition of multi-layer GNNs, collect all kinds of high order information step by step, into decoder with global evidence final decision with recent high approximation order parser, GNNs output calculation based on the previous layer GNN layer node, said and GNN node vector updates can check all in the middle of the tree, instead of just extracting the higher-order features of an intermediate tree, therefore, it can reduce the influence of subprime in the middle of the analytical results.

This parser significantly improves the performance of the baseline parser on long sentences, but slightly worse on short (length $< 10$) dependent lengths.

## 3 Text Matching

Text matching in chatbots is a critical step, matching algorithms must enhance the correlation between posts and responses (B. Hu, et al. 2014) [9].

### 3.1 Single-Turn Response Matching

#### 3.1.1 Traditional Matching Algorithm

Early matching techniques mostly matched at the lexical level, that is, how much the query field covers to calculate the matching score between the two. The higher the score, the better the matching degree of the query. The traditional matching models mainly include BoW, VSM, BM25, SimHash, etc. Among them, the most classic models are WMD, BM25. Matt et al. (2015) [7] associates

word embeddings with EMD to measure document distance. WMD (word mover 's distance) algorithm is proposed, it models the document distance as a combination of the semantic distances of the words in the two documents, such as the Euclidean distance of the word vectors corresponding to any two words in the two documents and then weighted sum. The BM25 algorithm is commonly used to search for correlation bisectors. It performs morpheme analysis on Query to generate morpheme qi; then, for each search result D, calculate the correlation score of each morpheme qi and D, and finally, weight the sum of the correlation scores of qi and D to obtain Correlation score between Query and D.

### 3.1.2 Deep Matching Algorithm

Matching algorithms are based on vocabulary matching, so they have great limitations. Deep learning methods can solve the problem of semantic limitations in traditional methods by extracting features and training data from the original data, and with the technology of word vector to achieve semantic level matching. In addition, based on the hierarchical structure of the neural network, the deep matching model can better establish a hierarchical matching model. Generally, deep text matching models are divided into two categories, representation model and interaction model.

Representation model focuses on the construction of the presentation layer, which transforms text into a unique overall representation vector at the presentation layer which is displayed in Fig. 2.

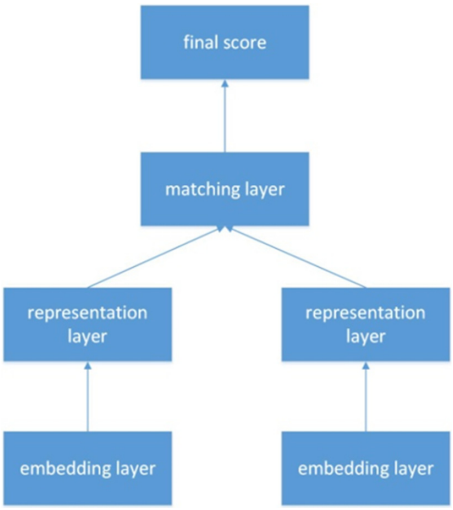


**Fig. 2.** Representation model framework.

The representational model is based on the Siamese network (Yann Lecun et al. 2005). The two texts are first mapped to a unified space, and the overall

semantics of the text are extracted before matching. At the matching layer, the dot product, cosine, Gaussian distance, MLP, similarity matrix and other methods are used for interactive calculation. Deep Structured Semantic Models (DSSM) (Huang PS et al. 2013) is a basic representational model. It first encodes two pieces of text into a fixed-length vector, and then calculates the similarity between the two vectors. The relationship between texts, where Q is a query and D is each candidate document.

The disadvantage of DSSM is that it uses the bag-of-words model (BoW), which loses word order information and context information, and it uses a weakly supervised, end-to-end model, with unpredictable prediction results. In response to the shortcomings of the DSSM bag-of-words model losing context information, CNN-DSSM (Shen, Yelong et al. 2014) [11] emerged at the historic moment. Its difference from DSSM mainly lies in the input layer and the presentation layer. CNN-DSSM uses CNN to extract local information, and then uses max pooling to extract and summarize global information in the upper layer. It can keep the context information more effectively, but cannot keep the context information that is far apart. To address this shortcoming, LSTM-DSSM (Palangi, Hamid et al. 2014) [12] was proposed. Here is its overall network structure (Fig. 3):

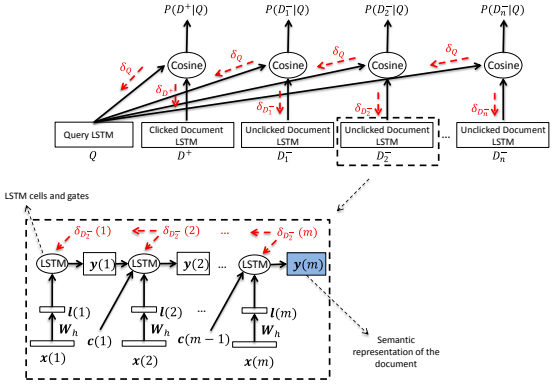


**Fig. 3.** LSTM-DSSM model.

In addition, other representational models including ARC-I (Hu, Baotian et al. 2015) [9], CNTN (Qiu X et al. 2015) [13], MultiGranRNN (Yin W et al. 2015) [14], and so on. Representational models can pre-process text. Constructing an index, but it will lose the semantic focus of the resulting sentence representation, are prone to semantic deviation, and it is difficult to measure the contextual importance of the word. Therefore, interaction model is proposed.

The interactive model discards the idea of post-matching. It assumes that the global matching degree depends on the local matching degree. The first matching between words is performed at the input layer, and the matching result is used as a grayscale image for subsequent modeling (Fig. 4):

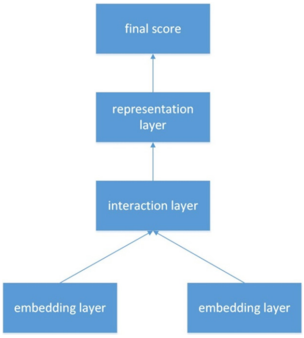


**Fig. 4.** Interaction model framework.

Interaction model uses the matching signal between words as a grayscale image, and then performs subsequent modeling abstraction. In the interaction layer, an interaction matrix is formed by two text words and words. In the representation layer, the interaction matrix is abstractly represented, using CNN or S-RNN (Yu, Zeping et al. 2018) [15]. Bao et al. (2014) [9] proposed the ARC-II network structure for the text matching model. Assuming that the length of both sentences is N and the embedding dimension is D, Then use a 3 * 3 convolution kernel to scan on an N * N picture, each scan 3 horizontal grids, 3 vertical grids, which respectively represent the words corresponding to two sentences, and then take 6 words, A total of 6 * D, then the size of the convolution kernel is also 6 * D. The convolution kernel and the selected word are multiplied and added (there is no activation function here), and finally a value is obtained, and the 3 * 3 volume is moved Kernel, and finally get a convolution picture (Fig. 5):

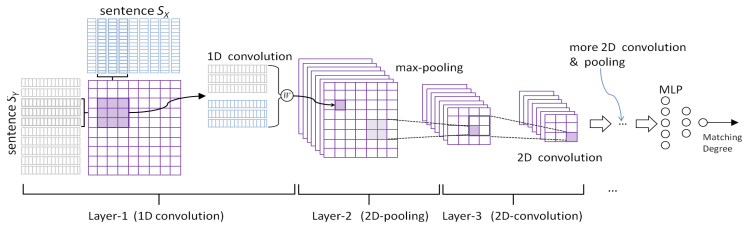


**Fig. 5.** ARC-II model.

In addition, Liang Pang et al. (2016) [16] proposed the MatchPyramid model, which uses image recognition to perform text matching, converts text matching to Text Matrix, and builds a CNN pyramid model to complete matching prediction. Construct matching matrices from three perspectives, consider the two-to-two relationship between words in sentences more carefully, construct three matrices for superposition, treat these matrices as pictures, and use convolutional neural networks to extract features from the matrices (Fig. 6).

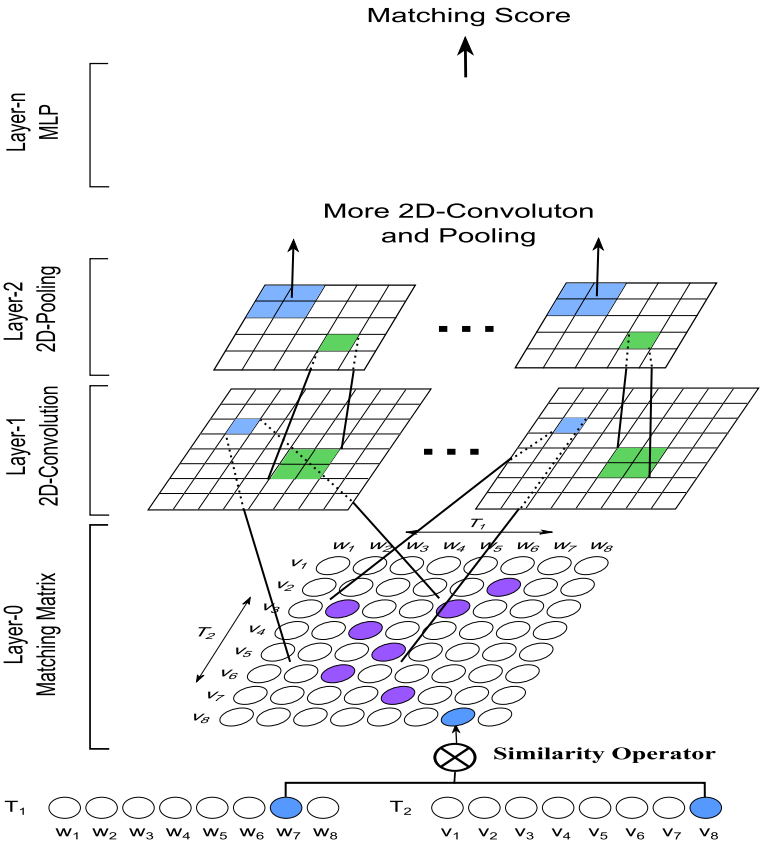


**Fig. 6.** MatchPyramid model.

Qian Chen et al. (2017) [17] proposed another text similarity calculation model ESIM, which consists of four parts: Input Encoding, Local Inference Modeling, Inference Composition and Prediction. Intra-sentence attention is mainly used to realize local inference, and further to achieve global inference. The Input Encoding layer uses BiLSTM for feature extraction and keeps the hidden state, where a and b represent the premise p and hypothesis h, and i and j represent different moments:

$$\bar{a_i} = BiLSTM(a, i), i \in [1, ..., l_a]$$

$$\bar{b_j} = BiLSTM(b, j), j \in [1, ..., l_b]$$

Next, calculate the similarity between the two sentences word to obtain a 2-dimensional similarity matrix, and then perform a local inference of the two sentences. Use the previously obtained similarity matrix to combine the a and

b sentences. Generate similarity-weighted sentences to each other with the same dimensions.

$$\widetilde{a_i} = \sum_{j=1}^{l_b} \frac{exp(e_{ij})}{\sum_{k=1}^{l_b} exp(e_{ik})} \bar{b_j}, i \in [1, ..., l_a]$$

$$\widetilde{b_j} = \sum_{i=1}^{l_a} \frac{exp(e_{ij})}{\sum_{k=1}^{l_a} exp(e_{kj})} \bar{a_i}, j \in [1, ..., l_b]$$

In the Inference Composition layer, the previous value is sent to the BiLSTM again to capture the local inference information and context for inference combination. Finally, the tanh activation function is used to send it to the Softmax layer.

Sentences should not only consider the direction from the question to the answer, but should also infer the question from the answer. Zhiguo Wang et al. (2017) [18] proposed the BiMPM model. Its innovation lies not only in the bidirectionality, but also in the consideration of sentences. There are 4 different ways to interact with each other, and then all the results are stitched and predicted. Word Representation Layer, representing each word in the sentence as a d-dimensional vector. Context Representation Layer fuses the context information into the representation of each time-step of P and Q. The output of the Matching Layer is two sequences. Each vector in the sequence is a certain time-step of a sentence matches all the time-steps of another sentence. Aggregation Layer aggregates two sequences of matching vectors into a fixed-length matching vector (Fig. 7).

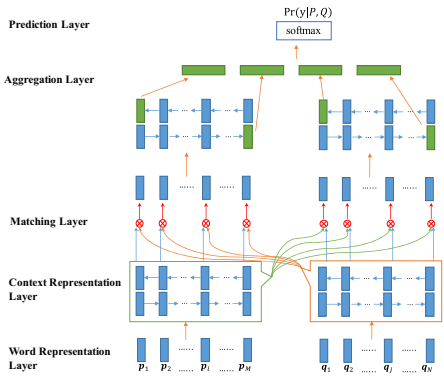


**Fig. 7.** BiMPM model.

In addition, typical interaction models also include DeepMatch, ABCNN (Wenpeng Yin et al. 2016) [19], DeepRank (Liang Pang et al. 2017) [20], IR-Transformer, and so on.

### 3.2 Multi-turn Response Matching

Interaction model can better grasp the semantic focus and better model the contextual importance, but it ignores global information such as syntax and cross-sentence comparison, and cannot describe global matching information from local matching information. And deep text matching is a single-round match question and answer for retrieval dialogues, which cannot complete multiple rounds of tasks. Therefore, based on a single round, adding other feature extraction techniques can complete multiple rounds of dialogue.

Xiangyang Zhou et al. (2016) [21] proposed to merge multiple rounds of question-and-answer sentences into one column, separated by _SOS_ at the connection, regarding the entire conversation history as "one sentence" to match the next sentence, and merge the entire conversation history into one column. After doing word embedding, lexical-level features are extracted through the GRU module and matched with candidate responses. In addition, the article proposes to match each text once, which means the combination of word-level and utterance-level. The loss function used in the integrated model is disagreement-loss (LD) and like-hood-loss (LL), it is an extension of a single-round question-and-answer representation model.

SMN was proposed by Yu Wu et al. (2017) [22], it is an extension of the single-turn Q & A interaction model. Constructing interactive representations of historical questions and answers and candidate replies is important feature information, so we use the matching matrix in semantic matching to construct a model by combining CNN and GRU. Here we consider the similarity matrix of two texts as an image, and then use the image classification model CNN to obtain higher level similarity feature representations (such as phrase level, segment level, etc.), and finally obtain the global similarity matching feature. Both the Multi-view and SMN models treat the conversation history as a whole. This will ignore the internal characteristics of the conversation history. For example, a conversation often includes multiple topics. In addition, the importance of words and sentences in a conversation is also different. Aiming at the information characteristics in these dialogue histories, a DUA model (Zhousheng Zhang et al. 2018) [23] was proposed. The author believes that the last sentence in the historical dialogue information is usually the most critical. Then make self-attention with utterance as the unit, the purpose is to filter redundancy (such as meaningless empty words) and extract key information. DUA focuses on multi-level feature extraction for both vocabulary and sentences, but in the case of many rounds, it will misjudge the correct response.

In order to solve the problem of misjudgment of response, the attention mechanism is applied to multiple rounds of dialogue. Xiangyang Zhou et al. (2018) [24] used self-attention and cross-attention to extract response and context features, which are mainly divided into Representation, Matching and Aggregation. In the matching stage, the DAM model has two matching matrices, which are the core part of the entire model. The first column Mself is called self-attention-match, and the second column Mcross is called cross-attention-match. Mself consists of Ui and R composed of L matrices obtained by Ui and response in each sentence

of the previous layer. Mcross uses two layers of traditional attention to calculate the alignment matrix.

Chongyang Tao et al. (2019) [25] adds a granularity of local information based on the two granularities of DAM, mainly in three forms: Word, Contextual, Attention, and adding word vectors to solve OOV problems. They did a lot of detailed experiments to compare the contribution of the three granularities and the impact of the rounds of conversation and the length of the utterance on the three granularities. The final conclusion is that Contextual contributes the most, and it performs better than Attention when there are few or many rounds.

## 4 Emotional Analysis

Emotional analysis is a relatively important part. Emotional analysis of the text focuses on analyzing users' emotional situation according to the content and context of the text, and making better responses and processing of the situation according to the results.

Aspect-based sentiment analysis (ABSA) provides more detailed information than general sentiment analysis because its purpose is to predict the emotional polarity of a given aspect or entity in a text. This is done by extracting the relevant aspects, called aspect terms, and detecting the emotional expression of each extracted aspect term, transforming it into an emotion classification at the aspect level. In the past, long-term short-term memory and attention mechanisms have been used to predict the emotional polarity of related objects, which is often complicated and requires more training time.

Neural networks have been widely used in affective analysis and sentence classification. Tree-based recursive neural Tensor Network (Socher et al. (2013) [26]) and Tree LSTM (tree-lstm, Tai et al. (2015) [27]) etc. all carry out syntactic interpretation of sentence structure, but these methods have problems such as low time efficiency and wrong parsing of review text. Recursive neural networks (RNNs) such as LSTM (Hochreiter and Schmidhuber et al. (1997)) and GRU (Chung et al. (2014)) have been used for the analysis of variable length data instances (Tang et al. (2015) [28]). There are also many models using convolutional neural networks (CNNs) (Collobert et al. (2011); Kalchbrenner et al. (2014) [29]; Kim et al. (2014); Conneau et al. (2016) [30]), which also proves that convolution can capture the complex structure of semantically rich text without tedious feature engineering.

Wei Xue, Tao Li et al. (2018) [31] summarized the previous method into two subtasks: aspect categorical emotion analysis (ACSA) and aspect term sentence analysis (ATSA), and proposed a new ACSA and ATSA model, namely gated convolutional network (GCAE) with directional embedding, which is more efficient and simpler than the model based on recursive network (Wang et al. (2016) [32]; Tang et al. (2016); Ma et al. (2017); Chen et al. (2017)) (Figs. 8 and 9).

The model has high precision and efficiency. First, this new gated unit can automatically output emotional characteristics based on a given aspect or entity, which is much simpler than the attention layer used in existing models. Secondly,

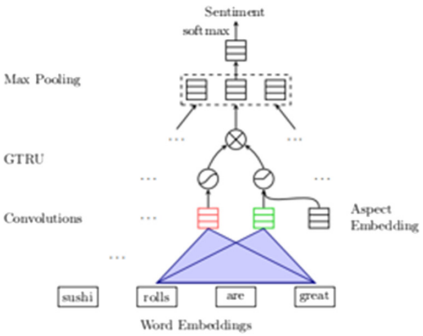


**Fig. 8.** GCAE for ACSA.

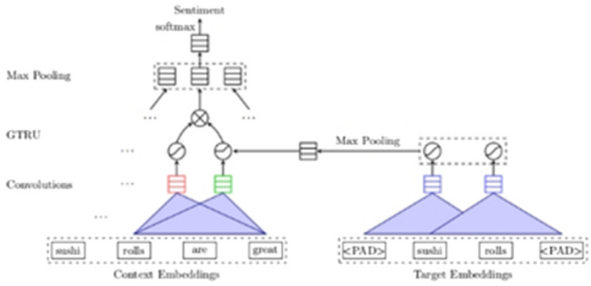


**Fig. 9.** GCAE for ACSA.

since the convolutional layer is not time-dependent like the LSTM layer, and the gating unit can work independently, the model calculation can be easily parallelized during the training process.

Ruidan He, Wee Sun Lee et al. (2019) [33] proposed an interactive multi-task learning network (IMN) for end-to-end aspect-based emotion analysis, for co-extraction of aspects and viewpoints, as well as emotion classification at the aspect level. It can solve two tasks at the same time, so that the interaction between the two tasks can be better used. In addition, IMN allows AE (aspect term extraction) and AS (aspect level affective classification) to be trained in related document-level tasks, utilizing knowledge from a larger document-level corpus. IMN introduces a new messaging mechanism that allows tasks to interact with each other. Specifically, it sends useful information from different tasks back to the potential Shared representation, and then combines this information with the Shared potential representation to provide all tasks for further processing. This is done iteratively, allowing information to be modified and propagated across multiple links as the number of iterations increases. Among them, a simple way to perform AE and AS simultaneously is multi-task learning, using a Shared network and two task-specific networks to derive a Shared feature space and two task-specific feature Spaces. Multi-task learning adopts

the Shared representation, and improves the generalization ability of the model under certain conditions by learning the correlation between tasks in parallel.

The traditional multi-task learning still does not explicitly simulate the interaction between tasks, and the interaction between the two tasks is only to promote learning behavior through false inferences, and this implicit interaction is uncontrollable. IMN not only allows for Shared representations, but also explicitly models interactions.

## 5 Conclusion

Researches relevant to the field of chatbot conversational systems have been developing rapidly. As the basic work in natural language processing, syntactic analysis has laid a solid foundation for NLP application scenarios such as semantic analysis and emotional expression. Text matching is an essential problem in conversational systems. How to choose a suitable text matching model for different tasks is an important challenge. As people's requirements for chatbots continue to increase, emotional analysis plays an increasingly important role. In this paper, we summarize various existing technologies behind different modules, and compare the advantages and disadvantages of each technology. The modular survey provides a preliminary understanding for beginners who are new to the field of conversational systems, and it is helpful for researchers in related fields to stitch and optimize models.

## References

1. Buchholz, S., Marsi, E.: CoNLL-X shared task on multilingual dependency parsing. In: Proceedings of the CoNLL 2006, pp. 149–164 (2006)
2. Chen, D., Manning, C.D.: A fast and accurate dependency parser using neural networks. In: EMNLP, pp. 740–750 (2014)
3. Weiss, D., Alberti, C., Collins, M., Petrov, S.: Structured training for neural network transition-based parsing. In: ACL (1), pp. 323–333 (2015)
4. Ma, M., Huang, L., Zhou, B., Xiang, B.: Dependency-based convolutional neural networks for sentence embedding. In: ACL (2), pp. 174–179 (2015)
5. Kim, Y.: Convolutional neural networks for sentence classification. In: EMNLP, pp. 1746–1751 (2014)
6. Ji, T., Wu, Y., Lan, M.: Graph-based dependency parsing with graph neural networks. In: ACL (1), pp. 2475–2485 (2019)
7. Kunsner, M.J., Sun, Y., Kolkin, N.I., Weinberger, K.Q.: From word embeddings to document distances. In: ICML, pp. 957–966 (2015)
8. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: CVPR (1), pp. 539–546 (2005)
9. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: NIPS, pp. 2042–2050 (2014)
10. Huang, P.S., He, X., Gao, J., et al.: Learning deep structured semantic models for web search using clickthrough data. In: CIKM, pp. 2333–2338 (2013)

11. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: A latent semantic model with convolutional-pooling structure for information retrieval. In: CIKM, pp. 101–110 (2014)
12. Palangi, H., et al.: Semantic modelling with long-short-term memory for information retrieval. CoRR abs/1412.6629 (2014)
13. Qiu, X., Huang, X.: Convolutional neural tensor network architecture for community-based question answering. In: IJCAI, pp. 1305–1311 (2015)
14. Yin, W., Schütze, H.: MultiGranCNN: an architecture for general matching of text chunks on multiple levels of granularity. In: ACL (1), pp. 63–73 (2015)
15. Yu, Z., Liu, G.: Sliced recurrent neural networks. In: COLING 2018, pp. 2953–2964 (2018)
16. Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., Cheng, X.: Text matching as image recognition. In: AAAI, pp. 2793–2799 (2016)
17. Chen, Q., Zhu, X., Ling, Z.-H., Wei, S., Jiang, H., Inkpen, D.: Enhanced LSTM for natural language inference. In: ACL (1), pp. 1657–1668 (2017)
18. Wang, Z., Hamza, W., Florian, R.: Bilateral multi-perspective matching for natural language sentences. In: IJCAI, pp. 4144–4150 (2017)
19. Yin, W., Schüutze, H., Xiang, B., Zhou, B.: ABCNN: attention-based convolutional neural network for modeling sentence pairs. In: TACL4, pp. 259–272 (2016)
20. Pang, L., Lan, Y., Guo, J., Xu, J., Xu, J., Cheng, X.: DeepRank: a new deep architecture for relevance ranking in information retrieval. In: CIKM, pp. 257–266 (2017)
21. Zhou, X., et al.: Multi-view response selection for human-computer conversation. In: EMNLP, pp. 372–381 (2016)
22. Wu, Y., Wu, W., Xing, C., Zhou, M., Li, Z.: Sequential matching network: a new architecture for multi-turn response selection in retrieval-based chatbots. In: ACL (1), pp. 496–505 (2017)
23. Zhang, Z., Li, J., Zhu, P., Zhao, H., Liu, G.: Modeling multi-turn conversation with deep utterance aggregation. In: COLING, pp. 3740–3752 (2018)
24. Zhou, X., et al.: Multi-turn response selection for chatbots with deep attention matching network. In: ACL (1), pp. 1118–1127 (2018)
25. Tao, C., Wu, W., Xu, C., Hu, W., Zhao, D., Yan, R: Multi-representation fusion network for multi-turn response selection in retrieval-based chatbots. In: WSDM, pp. 267–275 (2019)
26. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J.: Recursive deep models for semantic compositionality over a sentiment Treebank. In: EMNLP, pp. 1631–1642 (2013)
27. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. In: ACL, pp. 1556–1566 (2015)
28. Tang, D., Qin, B., Feng, X., Liu, T.: Effective LSTMs for target-dependent sentiment classification. In: COLING, pp. 3298–3307 (2016)
29. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. In: ACL, pp. 655–665 (2014)
30. Conneau, A., Schwenk, H., Barrault, L., LeCun, Y.: Very deep convolutional networks for text classification. In: EACL, pp. 1107–1116 (2016)
31. Xue, W., Li, T.: Aspect based sentiment analysis with gated convolutional networks. In: ACL (1), pp. 2514–2523 (2018)
32. Wang, B., Liu, K., Zhao, J.: Inner attention based recurrent neural networks for answer selection. In: ACL (1) (2016)
33. He, R., Lee, W.S., Ng, H.T., Dahlmeier, D.: An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In: ACL (1), pp. 504–515 (2019)

34. Xu, J., Sun, X.: Dependency-based gated recursive neural network for Chinese word segmentation. In: ACL (2) (2016)
35. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. Int. J. Comput. Vis. **40**(2), 99–121 (2000)
36. Norouzi, M., Fleet, D.J., Salakhutdinov, R.: Hamming distance metric learning. In: NIPS, pp. 1070–1078 (2012)