DEMN: Distilled-Exposition Enhanced Matching Network for Story Comprehension

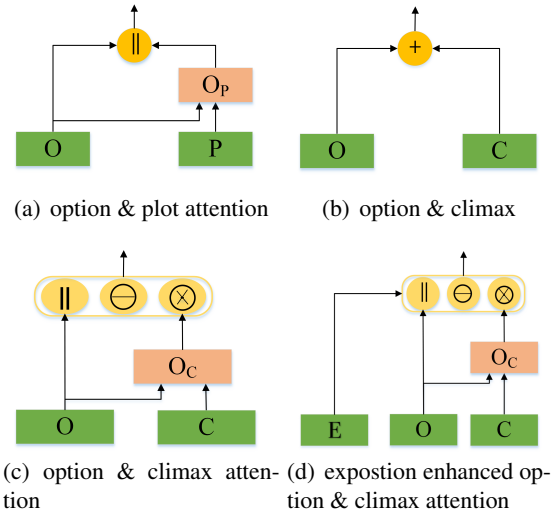
Chunhua Liu[1] **Haiou Zhang**[1] **Shan Jiang**[1] **Dong Yu**[1,2] ✉
[1] Beijing Language and Culture University
[2] Beijing Advanced Innovation for Language Resources of BLCU
{chunhualiu596, jiangshan727}@gmail.com
{hozhangel, yudong_blcu}@126.com

Abstract

This paper proposes a Distilled-Exposition Enhanced Matching Network (DEMN) for story-cloze test, which is still a challenging task in story comprehension. We divide a complete story into three narrative segments: an *exposition*, a *climax*, and an *ending*. The model consists of three modules: input module, matching module, and distillation module. The input module provides semantic representations for the three segments and then feeds them into the other two modules. The matching module collects interaction features between the ending and the climax. The distillation module distills the crucial semantic information in the exposition and infuses it into the matching module in two different ways. We evaluate our single and ensemble model on ROCStories Corpus (Mostafazadeh et al., 2016), achieving an accuracy of 80.1% and 81.2% on the test set respectively. The experimental results demonstrate that our DEMN model achieves a state-of-the-art performance.

1 Introduction

Story comprehension is a fascinating task in natural language understanding with a long history (Jones, 1974; Turner, 1994). The difficulty of this task arises from the necessity of commonsense knowledge, cross-sentence reasoning, and causal reasoning between events. The recently emerged story-cloze test (Mostafazadeh et al., 2016) focuses on commonsense story comprehension, which aims at choosing the most plausible ending from two options for a four-sentence story (also called plot).



(a) option & plot attention (b) option & climax

(c) option & climax attention (d) expostion enhanced option & climax attention

Figure 1: Four strategies of modeling a story

Recently, methods based on linear classifier with handcrafted features, as well as neural network (NN), have been proposed for the story-cloze test. Handcrafted features based methods (Chaturvedi et al., 2017; Mostafazadeh et al., 2016; Schwartz et al., 2017) extract commonsense knowledge like events sequence and sentiments trajectory by external tools to help the story understanding. first Among NN based methods, the val-LS-skip model (Srinivasan et al., 2018) represents the last sentence in the plot and the option ending with skip-embeddings, and then processes them with a simple feed-forward neural network. Their experiments show that representing the whole four-sentence plot with the ending performs worse than only using the fourth sentence with the ending.

However, intuitively speaking, a coherent ending should be related to the whole plot, instead of just the fourth sentence. This intuition is opposite to the conclusion of Srinivasan et al. (2018). To explore whether the content in a plot, except for only the fourth sentence, can assist in choosing a correct ending, we observed a large number of stories and discovered two phenomena: *(1) The ending is usually directly affected by the last sentence in a plot. (2) The first three sentences in a plot usually provide background settings about the character, time, and place of the story, which influences the ending implicitly but essentially.* Inspired by our findings, we divide a complete story into three parts: an *exposition part* (the first three sentences), a *climax part* (the fourth sentence), and an *ending*. Within a story, exposition is the beginning of the narrative arc. It introduces key scenarios, themes, characters, or settings about a story, and creates the rising action of the story which then reaches the climax and continues through into the resolution. In addition, we still denote the four sentences as a plot. Table 1 shows an example from the ROCStories Corpus, consisting of the three segments we described above.

Based on the narrative segments of a story, we summarize four strategies for the story-cloze test task. We show the four strategies in Figure 1, and here we denote the ending as the option. The strategy (a) Cai et al. (2017) treats the exposition part and the climax as a whole. The strategy (b) just considers the representations of the climax and the ending without interaction (Srinivasan et al., 2018). The strategy (c) interacts the ending with the climax in word level to acquire more sufficient information. The strategy (d) uses a distilled exposition to enhance the interaction between the climax and the ending. Previous studies highlight the importance of the climax but ignore the importance of the exposition. How to exploit useful information from the exposition, and further help the model to understand the story is the key problem we are trying to figure out in this work.

Following the strategy (c) and (d), we propose a Distilled-Exposition Enhanced Matching Network (DEMN) for the story-cloze test. The model comprises three modules: an input module, a matching module, and a distillation module. The input module is constructed by an embedding layer with various embeddings and an encoding layer with a BiLSTM. The matching module matches the option ending with the climax, using a matching network proposed by (Liu et al., 2018a). And the distillation module focuses on distilling the exposition and injects it to the matching network. Our model does not only match the climax with the ending explicitly but also takes full advantage of the exposition.

Exposition: Tom was studying for the big test. He then fell asleep do to boredom. He slept for five hours.
Climax: He woke up shocked.
False-Option: Tom felt prepared for the test.
True-Option: Tom hurried to study as much as possible before the test.

Table 1: An example from the ROCStories Corpus.

We conduct experiments on the ROCStories Corpus (Mostafazadeh et al., 2016). Our model achieves an accuracy of 80.1% on the story-cloze test, outperforming all previous methods. Our key contributions are as follows:

- We divide a story into three narrative segments: an exposition, a climax, and an ending.
- We use a matching network to model the interaction between the climax and ending explicitly.
- We distill the crucial parts in the exposition to help identify a coherent ending, which is proved to be significantly effective.

2 Model

2.1 Model Overview

The overview of our DEMN model is shown in Figure 2. We denote $e = \{e_1, e_2, \cdots, e_{|e|}\}$ as the exposition, $c = \{c_1, c_2, \cdots, c_{|c|}\}$ as the climax and $o = \{o_1, o_2, \cdots, o_{|o|}\}$ as one of the option endings.

Input Module: Embedding Layer This layer aims to map each word in e, c, o to a semantically rich d-dimensional embedding. Following (Chen et al., 2017a), we use various embeddings to construct the d-dimensional embedding, including the pre-trained 300-dimensional Glove word embedding, the part-of-speech (POS) embedding, named entity recognition (NER) embedding, term frequency (TF) feature and exact-match feature. Furthermore, we

use the relation embedding (Rel). For each word, the relationship with any other word with another sequence will be recorded.

Input Module: Encoding Layer The goal of this layer is to acquire the context representation of the exposition, the climax, and the option. A single-layer bidirectional LSTM (BiLSTM) (Hochreiter and Schmidhuber, 1997) is applied to transform their word embeddings respectively. Then, we can obtain the exposition hidden outputs $\bar{e} = \text{BiLSTM}(e) \in \mathbb{R}^{|e| \times 2d}$, the climax hidden outputs $\bar{c} = \text{BiLSTM}(c) \in \mathbb{R}^{|c| \times 2d}$, and the option hidden outputs $\bar{o} = \text{BiLSTM}(o) \in \mathbb{R}^{|o| \times 2d}$.

Matching Module This module is responsible for matching the option $\bar{o}$ with the climax $\bar{c}$. It first computes word level attention vectors between the $\bar{c}$ and the $\bar{o}$. Then the attention vectors, along with the hidden outputs from the encoding layer, are matched at word-level and are further aggregated in multi-turn with the memory component. Finally, max pooling and average pooling are applied on the aggregation outputs to form a fixed length aggregation vector for the output layer.

Distillation Module The purpose of this module is to distill the exposition and infuse it to the matching process. We first present how to distill the crucial information in the exposition, and then inject the information in two different fashions to enhance the whole matching process.

Output Layer This layer is used to predict which candidate ending is more reasonable. The input of this layer is the aggregation vector from the matching process. We apply a two-layer FNN with $tanh$ activations as a score function to produce the final prediction label.

2.2 Matching Module

In this layer, we first use word-level attention to model interactions among the climax $\bar{c}$ and the ending $\bar{o}$. Then we use the multi-turn matching mechanism to compare the interactive representations $\tilde{o}^c$ with the original hidden outputs $\bar{o}$ (Liu et al., 2018a).

Sequence attention Here, we adopt dot product attention to model the interaction between two sequences. Given two d-dimensional vector sequences $x = \{x_1, x_2, \cdots, x_{|x|}\}$ and $y = \{y_1, y_2, \cdots, y_{|y|}\}$ with length $|x|$ and $|y|$ respectively, we define a sequence attention function $Attn(x, y)$ to compute the x-aware y representation as follows:

$$Attn(x, y) = \{\beta_i^T y\}_{i=1}^{|x|} \quad (1)$$

$$\beta_i = Softmax(x_i\, y^T) \quad (2)$$

where the $\beta_i \in \mathbb{R}^{1 \times |y|}$ indicates how x_i is relevant to each element of y.

Option-aware climax For each embedding $\bar{o}_i$, to find the related parts in the climax, we compare it with the hidden outputs $\bar{c}$ to obtain the option-aware climax representation $\tilde{o}^c$, where $\tilde{o}^c = Attn(\bar{o}, \bar{c})$. For each word in the option, the relevant content in the climax will be selected and fused into $\tilde{o}_i^c$.

Matching option with climax In order to better infer whether the option ending is semantically consistent with the climax, we use the following three matching functions to compare the $\bar{o}$ and the $\tilde{o}^c$ (Wang and Jiang, 2016; Wang et al., 2018) :

$$u^1 = \text{ReLu}(W^1(\bar{o} \,||\, \tilde{o}^c) + b^1) \quad (3)$$

$$u^2 = \text{ReLu}(W^2(\bar{o} \ominus \tilde{o}^c) + b^2) \quad (4)$$

$$u^3 = \text{ReLu}(W^3(\bar{o} \otimes \tilde{o}^c) + b^3) \quad (5)$$

where $||$, $\ominus$, and $\otimes$ represent the concatenation, element-wise subtraction, and element-wise multiplication between two matrices respectively. These operations can match the ending with the climax from different views.

Multi-turn aggregation In this layer we aim to integrate the matching matrices $\{u_i\}_{i=1}^3$ to acquire deeper understanding about the relationship between the option and the climax. Following (Liu et al., 2018a), we utilize another BiLSTM with an external memory matrix to aggregate the these matching matrices.

$$h^t = \text{BiLSTM}(W_h(u^t \,||\, m^{(t-1)})) \quad (6)$$

$$m^t = g^t \otimes h^t + (1 - g^t) \otimes m^{(t-1)} \quad (7)$$

$$g^t = \sigma(W_g(h^t \,||\, m^{(t-1)}) + b_g) \quad (8)$$

where W_h, W_g, and b_g are parameters to be learned, σ is a sigmoid function, and $m^{(t-1)}$ is a memory vector that stores the history aggregation information.

The last memory matrix m^3 stores the whole matching and aggregation of information. To obtain a global aggregation representation, we convert it to a fixed length vector with max and average pooling. The final aggregation vector $\hat{o} =$

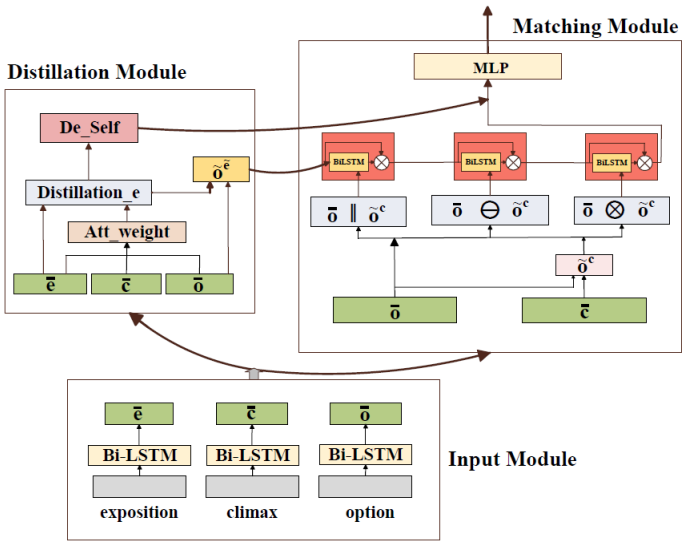


Figure 2: Overview of our DEMN model.

$MaxPooling(m^3) \,||\, AvePooling(m^3)$.

2.3 Distillation Module

In this subsection, we focus on distilling the exposition and incorporating the distilled exposition into the matching process.

Exposition distillation In a four-sentence plot, the exposition generally contains abundant background information of a story. Along with the development of a story, part of the background information may become useless and noisy. To avoid the negative effect caused by redundant content, this module aims to distill the exposition to maintain the vital content and filter out the irrelevant content. The distillation process can be divided into two steps.

The first step chooses the relevant context of the climax and the ending, by computing attention with the exposition. We compute the exposition-aware climax by $\tilde{e}^c = Attn(\bar{e}, \bar{c})$. Similarly, we get the exposition-aware ending $\tilde{e}^o = Attn(\bar{e}, \bar{o})$.

The second step distills the the exposition using a carefully designed attention weight.

$$s = (\bar{e} \ominus \tilde{e}^c) \otimes (\bar{e} \ominus \tilde{e}^o) \tag{9}$$

$$\alpha = softmax(s^T s) \tag{10}$$

$$\tilde{e} = \alpha \bar{e} \tag{11}$$

The attention weight α embodies the climax and option which are carefully selected. The $\tilde{e}$ is called the distilled exposition, which is obtained by distilling the exposition, the climax, and the option. The distillation process experienced multiple information selection. In this way, the crucial parts in the exposition that related to the climax and the option can be highlighted.

DEEM: Distilled-Exposition Enhanced Memory The distilled exposition $\tilde{e}$ organizes the relevant information about the climax and the ending, which can be used to enhance the matching process between the climax and the ending. To make the background information flow through each turn, we infuse the refined exposition to the initial memory component. We first compute the option-aware distilled exposition $\tilde{o}^{\tilde{e}} = Attn(\bar{o}, \tilde{e})$, then we use the $\tilde{o}^{\tilde{e}}$ to initialize the matching memory described in multi-turn aggregation:

$$m^0 = \tilde{o}^{\tilde{e}} \tag{12}$$

DEEAV: Distilled-Exposition Enhanced Aggregation Vector Word-level attended exposition can capture the relevant information in a particular view of the other word. However this lacks an overall representation about the distilled exposition itself. Summarizing all the exposition with different weights can provide the whole picture about the background settings. Hence, we first transform the exposition to a fixed-length vector with self-attention (Yang et al., 2016).

$$\hat{e} = \sum_{k=1}^{|e|} \alpha \tilde{e}, \qquad \alpha_i = Softmax(W\tilde{e}) \tag{13}$$

where the $\hat{e} \in \mathbb{R}^{2d}$ is a distilled exposition vector

that summarizes all the information of the exposition according to different important degrees.

Then, we combine the $\hat{e}$ with the output of the matching module together for final prediction.

$$v = \hat{o} \,||\, \hat{e} \tag{14}$$

where v does not only contain the interactive representation between the climax, but also holds the whole picture about the exposition. When the distilled-exposition vector $\hat{e}$ is used, the vector v is fed into the output layer to compute the probability of being a coherent option ending.

3 Experiments

3.1 Experimental Settings

Dataset We employ the ROCStories Corpus released by Mostafazadeh et al. (2016) to evaluate our model. There are 100k stories in the training set and 1871 stories in both validation set and test set. The training set contains stories with a plot and a correct option ending. The validation set and the test set both contain a plot and two option endings, including a correct one and an incorrect one.

Following previous studies, we train only on the validation set and evaluate on the test set. During training, we hold out 1/10 of the validation set to fine tune parameters, and save the best performing parameters for testing. For data preprocessing, we use spaCy [1] for sentence tokenization, Part-of-Speech tagging, and Name Entity Reorganization. The relations between two words are generated by ConceptNet [2].

Model Configuration

We implement our model with Pytorch [3]. We initialize the word embeddings by the pre-trained 300D GloVe 840B embeddings (Pennington et al., 2014) and keep them fixed during training. The word embeddings of the out-of-vocabulary words are randomly initialized. We use Adam (Kingma and Ba, 2015) for optimization. As for hyper-parameters, we set the batch size as 64, the learning rate as 0.008, the dimension of BiLSTM and the hidden layer of MLP as 96, the L2 regularization weight decay coefficient as 3e-8, the dropout rate for word embedding and the initial memory in multi-turn aggregation as 0.4 and 0.41 respectively. The dimension of POS embedding, NER embedding, and Rel embedding are set as 18, 8, and 10 respectively. They all are fine-tuned during the training.

[1] https://github.com/explosion/spaCy
[2] http://conceptnet.io/
[3] http://pytorch.org/

Models	Test-Acc
DSSM (Mostafazadeh et al., 2016)	58.5%
HIER&ATT (Cai et al., 2017)	74.7 %
MSAP (Schwartz et al., 2017)	75.2%
val-LS-skip (Srinivasan et al., 2018)	76.5%
HCM (Chaturvedi et al., 2017)	77.6%
Memory Chains (Liu et al., 2018b)	78.5%
option-climax (single)	**77.8%**
DEMN (single)	**80.1%**
DEMN (ensemble)	**81.2%**

Table 2: Test accuracy of the SOA models.

3.2 Results and Analysis

Baselines Table 2 shows the results of our DEMN model along with the published models on this dataset. The DSSM model is reported in Mostafazadeh et al. (2016), which applies an LSTM to transform the four-sentence plot and the option to corresponding sentence vectors. Based on these vectors, the model chooses the option that has higher cosine similarity with the plot. Both the HIER&ATT and MSAP show that using ending alone can get better performance. The HIER&ATT model uses hierarchical BiLSTM with attention to encode the plot and the ending. The MSAP model trains a linear classifier model with stylistic features and language model predictions. The val-LS-skip model simply sums the skip-embeddings of the climax and the ending for final prediction. Both the HCM model and the Memory Chains model exploit three aspects of semantic knowledge, including the event sequence, sentiment trajectory, and topic consistency to model the plot and the ending. However, they use these features differently. The HCM model designs hidden variables to weight the three aspects, while the Memory Chains model leverages a neural network with memory chains to learn representation for each aspect.

DEMN Results Analysis The option-climax model only uses the climax to match with the option.

distillation	Models	Test-Acc
	Full Model	80.1%
distillation	w/o exposition-mem	78.5%
exposition	w/o exposition-vec	78.8%
w/o	exposition-mem&vec	79.3%
distillation	w/o exposition-mem	78.8%
exposition	w/o exposition-vec	78.6%

Table 3: Ablations studies about the influence of distilling exposition.

Models	Test-Acc
Full Model	80.1%
w/o exposition-aware climax	79.7%
w/o exposition-aware option	78.9%
w/o exposition-aware climax&option	78.6 %

Table 4: Ablation studies about different ways of distilling exposition.

This model reaches an accuracy of 77.8%, which provides a strong baseline for infusing the exposition into the interaction process. Our DEMN model achieves an accuracy of 80.1%, outperforming the current state-of-the-art result (Liu et al., 2018b) with 1.6% absolute improvement. Our model exceeds the HIER&ATT by 5.4% in terms of accuracy. We attribute this improvement to separately handling the exposition and the climax, and the multiple word-level attentions. Comparing with the val-LS-skip model, our DEMN model yields 3.6% improvement, which proves that the exposition can promote the model effectively instead of demoting. This conclusion is exactly opposite to Srinivasan et al. (2018).

3.3 Ablation Study

To evaluate how different components contribute to the model performance, we design two groups of experiments. We will discuss the influence of distilling the exposition and different ways of distillation.

Influence of distilling exposition We conduct ablation study on two different ways of using the distilled exposition, by removing the distilled-exposition memory and distilled-exposition vector. Furthermore, to verify the effectiveness of the distillation, we design another three experiments without the distillation process. In this case, we use the original BILSTM hidden outputs of exposition $\bar{e}$ to replace the distilled exposition $\tilde{e}$. Table 3 shows the experimental results.

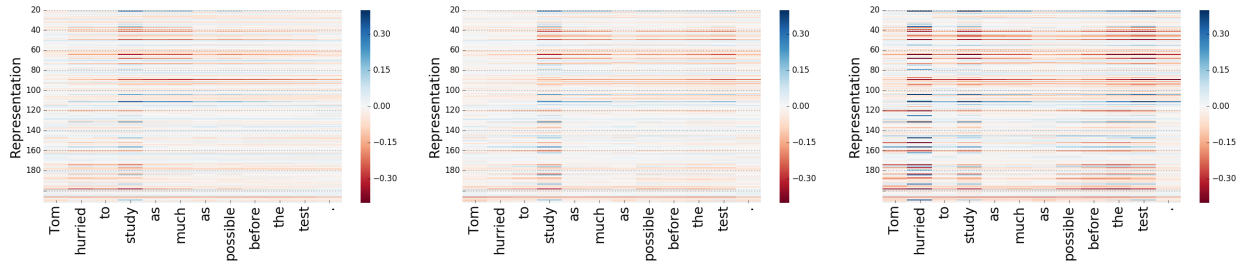
The first group of results in Table 3 shows the performance of removing two kinds of exposition separately. We observe a substantial drop, when we replace the distilled-exposition memory with zero memory. For each word in the option, the corresponding distilled-exposition memory can provide the different background knowledge about the story. This result proves that the information in the exposition is vital for choosing a correct option. On the other hand, removing the distilled-exposition vector impairs the performance as well. Compared with removing exposition memory, the accuracy declines less when removing exposition vector, indicating the former offers more benefits to the model.

The second group of results in Table 3 report the performance without distillation. We observed that the highest accuracy 79.3% appears when two kinds of exposition representation are incorporated into the model. Once again, these results show that the content in the exposition is helpful in choosing a coherent option-ending. We can also see that the models would be impaired, no matter which component is removed. However, the accuracy drops slightly than the first group (1.6% absolute vs 0.5% absolute, and 1.3% absolute vs 0.7% absolute).

Different ways of distilling exposition To investigate the influence of the attention weight to distill the exposition, we conduct an ablation study on the way of calculating the attention scores. In our DEMN model, the attention weights imply three aspects of information, including exposition itself, the exposition-aware climax, and the exposition-aware option. We observe that the accuracy is slightly influenced without the exposition-aware climax. While removing the exposition-aware option affects the performance more obviously. The degrees of performance degradation are various. Removing both of the interactive representations is the most detrimental. These reflect that distilling more accurate exposition is important.

3.4 Discussion

In this work, we adopt three matching features to exploit the relation between two sequences in word



(a) memory1 (b) memory2 (c) memory3

Figure 3: The memory representations in three turns

level, which plays an important role in matching network. To discuss how these matching features influence the model, we design a series of experiments and give further analysis on the learning curves of the test set.

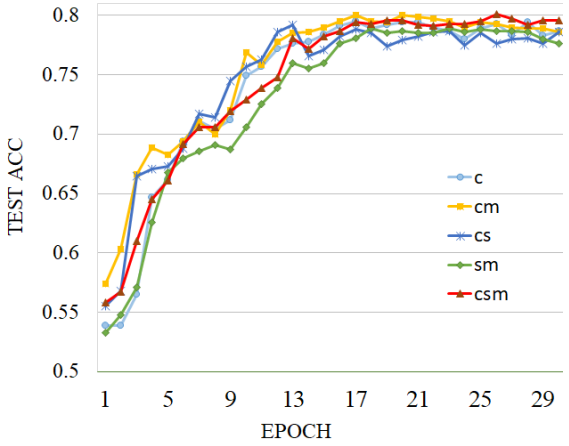


Figure 4: Learning curves of different matching features

Figure 4 presents the performance of using different matching features. Both curves of the "cs" and "cm" rise obviously in the early training stage, however, the "cs" suffer a sustained decrease after the peak. The two curves of "cm" and "csm" are very close in the latter training stage. Until the epoch 26, the "csm" reaches an accuracy of 80.1% on the test set with an accuracy of 81.8% on the validation set. We finally choose the model with "csm", hence it performs better on the validation set.

3.5 Visualization

To gain an insight into the multi-turn aggregation process employed by the model, we observed many visualized pictures of the memory representations during the multi-turn process of different examples. There is an obvious conclusion. From the pictures of memory1 to memory2, then to memory3: the colors of the keyword parts of the pictures are more and more prominent and obvious. Take the case in Table 1 for example. We can see that in Figure 3, the key word "study" is the captured in the memory1 and held its importance in the next two turns. In the memory2, we can see that "the test" is marked. As both "study" and "the test" are directly relevant to the topic described in the exposition. The most surprising thing is that the model pays attention to the word "hurried" in the final turn, which can be highly linked to the rest parts of scenario. In a summary, the important words are highlighted step by step along the multi-turn process.

3.6 Error analysis

In order to analyze which kind of problems can not be solved by our model, we observe some error cases. We find that it is difficult for the model to choose the right ending when the plot have lots of negation words, complicated phrases, or unrelated noisy words. Table 5 shows two error cases selected randomly. In the first case, we observe that the ending cannot be chose correctly only based on the climax. Taking the whole plot into consideration, the correct is obvious to us. However, because the model misunderstands the advanced phrase "write-up" and unable to capture sufficient information from the exposition, the score of false option is much higher than the true option. In the second case, the two endings cover too many same words and their prediction scores are quite close. We observe that

Exposition: Collin likes to dress up. One Halloween he decided to wear his costume to the office. Collin's boss did not permit costumes to be worn in the workplace. **Climax:** He received a write-up. **False-Option:** Collin received a raise and a promotion. **True-Option:** Collin was upset with his boss.
Exposition: Amy was visiting her best friend in Phoenix. It was her first time there. She was excited to see the town. **Climax:** She exited the airport and was struck by the heat. **False-Option:** Amy began shivering. **True-Option:** Amy began to sweat.

Table 5: Error cases

the word "heat" is the most crucial factor of the right choice and it occurs only once. The model was puzzled by other misleading words like "excited". So it is very challenging for the model to fully filter all the redundant information.

4 Related Work

The Story Cloze Test (Mostafazadeh et al., 2016) is proposed to evaluate the story understanding ability of a system. Recently, several neural networks are used to tackle this task. **HIER&ATT** (Cai et al., 2017) employs LSTM units to hierarchically encode both the ending and the ending-aware full context. A surprising finding is that only relying on the ending can achieve an accuracy of 72.7%. **Val-LS-skip** (Srinivasan et al., 2018) achieves a competitive result by using a single feed-forward neural network with pre-trained skip-embeddings of the last sentence and the ending. Their experiments show that the performance of using the whole plot is worse than just using the forth sentence. However, our DEMN model can extract useful information from the exposition part, which actually improves the accuracy rather than decreases it.

To explore the external knowledge to help the story understanding, three semantic aspects are frequently used, including events sequence, sentiment trajectory, and topical consistency (Lin et al., 2017; Chaturvedi et al., 2017; Liu et al., 2018b). The current state-of-the-art model **Memory Chains** (Liu et al., 2018b) adopts the EntNet (Henaff et al., 2017) to track the three semantic aspects of the full context with external neural memory chains.

Another closely related work is the matching network (Chen et al., 2017b; Wang and Jiang, 2016; Liu et al., 2018a), which is commonly used in natural language inference (MacCartney, 2009). The matching network can match the interactions between two sequences effectively. Among them, the **MIMN** model proposed by (Liu et al., 2018a) introduce a multi-turn inference with memory mechanism to compute three heuristic matching (Mou et al., 2016) representation between two sequences iteratively. Multi-turn inference can capture more detailed content about the interactions between two inputs, and it performs well on small-scale datasets. We use the multi-turn inference method to model the interactions between the climax and the ending. Furthermore, this paper focuses on exploiting the available information in the exposition to assist the interactions of the climax and the ending.

5 Conclusion

In this paper, we propose the Distilled-Exposition Enhanced Matching Network model for the story-close task. Our model achieves an accuracy of 80.1% on ROCStories Corpus, outperforming the current state-of-the-art model. In our task, we divide the story into an exposition, a climax, and an ending. The experimental result shows that matching the ending with the climax can achieve a strong baseline. This indicates that the interaction between the climax and the option is necessary. Further, we propose a method of distilling the exposition with the evidence provided by the climax and the ending. More specifically, we integrate the distilled exposition into the matching process in two ingenious manners, and yield a significant improvement.

Acknowledgements

This work is funded by Beijing Advanced Innovation for Language Resources of BLCU (TYR17001J), and The National Social Science Fund of China (16AYY007).

References

[Cai et al.2017] Zheng Cai, Lifu Tu, and Kevin Gimpel. 2017. Pay attention to the ending: Strong neural baselines for the ROC story cloze task. In *ACL*, pages 616–622.

[Chaturvedi et al.2017] Snigdha Chaturvedi, Haoruo Peng, and Dan Roth. 2017. Story comprehension for predicting what happens next. In *EMNLP*.

[Chen et al.2017a] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017a. Reading Wikipedia to answer open-domain questions. In *ACL*.

[Chen et al.2017b] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017b. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668. Association for Computational Linguistics.

[Henaff et al.2017] Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann LeCun. 2017. Tracking the world state with recurrent entity networks. *ICLR 2017*.

[Hochreiter and Schmidhuber1997] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. In *Neural Comput.*, volume 9, pages 1735–1780, Cambridge, MA, USA, November.

[Jones1974] Karen Spärck Jones. 1974. "understanding natural language, " by t. winograd (book review). *International Journal of Man-Machine Studies*, 6:279–281.

[Kingma and Ba2015] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

[Lin et al.2017] Hongyu Lin, Le Sun, and Xianpei Han. 2017. Reasoning with heterogeneous knowledge for commonsense machine comprehension. In *EMNLP*.

[Liu et al.2018a] Chunhua Liu, Shan Jiang, Hainan Yu, and Dong Yu. 2018a. Multi-turn inference matching network for natural language inference. *Natural Language Processing and Chinese Computing (NLPCC)*.

[Liu et al.2018b] Fei Liu, Trevor Cohn, and Timothy Baldwin. 2018b. Narrative modeling with memory chains and semantic supervision. In *ACL*, pages 1–10, Melbourne, Australia.

[MacCartney2009] Bill MacCartney. 2009. Natural language inference. *Ph.D. thesis, Stanford University*.

[Mostafazadeh et al.2016] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *HLT-NAACL*.

[Mou et al.2016] Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 130–136, Berlin, Germany, August. Association for Computational Linguistics.

[Pennington et al.2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

[Schwartz et al.2017] Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. 2017. The effect of different writing tasks on linguistic style: A case study of the roc story cloze task. In *CoNLL*.

[Srinivasan et al.2018] Siddarth Srinivasan, Richa Arora, and Mark O. Riedl. 2018. A simple and effective approach to the story cloze test. In *NAACL-HLT*.

[Turner1994] S.R. Turner. 1994. *The Creative Process: A Computer Model of Storytelling and Creativity*. L. Erlbaum.

[Wang and Jiang2016] Shuohang Wang and Jing Jiang. 2016. A compare-aggregate model for matching text sequences. In *ICLR2016*, volume abs/1611.01747.

[Wang et al.2018] Shuohang Wang, Mo Yu, Shiyu Chang, and Jing Jiang. 2018. A co-matching model for multi-choice reading comprehension. In *ACL*.

[Yang et al.2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *HLT-NAACL*.