Incorporating Structured Commonsense Knowledge in Story Completion

Jiaao Chen,[*1] **Jianshu Chen,**[2] **Zhou Yu**[3]
[1]Zhejiang University, [2]Tencent AI Lab
[3]University of California, Davis
3150105589@zju.edu.cn, jianshuchen@tencent.com,joyu@ucdavis.edu

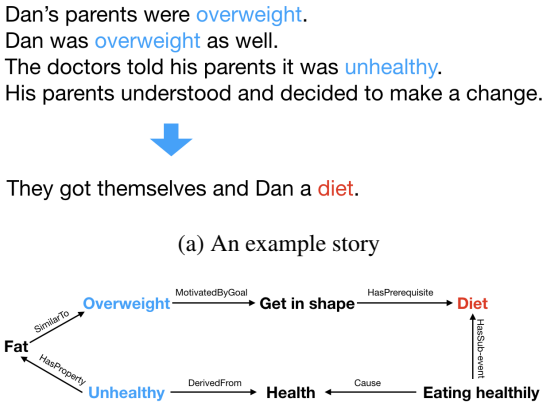
Abstract

The ability to select an appropriate story ending is the first step towards perfect narrative comprehension. Story ending prediction requires not only the explicit clues within the context, but also the implicit knowledge (such as commonsense) to construct a reasonable and consistent story. However, most previous approaches do not explicitly use background commonsense knowledge. We present a neural story ending selection model that integrates three types of information: narrative sequence, sentiment evolution and commonsense knowledge. Experiments show that our model outperforms state-of-the-art approaches on a public dataset, ROCStory Cloze Task (Mostafazadeh et al. 2017), and the performance gain from adding the additional commonsense knowledge is significant.

Introduction

Narrative is a fundamental form of representation in human language and culture. Stories connect individuals and deliver experience, emotions and knowledge. Narrative comprehension has attracted long-standing interests in natural language processing (Schubert and Hwang 2000), and is widely applicable to areas such as content creation. Enabling machines to understand narrative is also an important first step towards real intelligence. Previous studies on narrative comprehension include character roles identification (Valls-Vargas, Ontañón, and Zhu 2015), narratives schema construction (Chambers and Jurafsky 2009), and plot pattern identification (Jockers 2013). However, their main focus is on analyzing the stories themselves. In contrast, we concentrate on training machines to predict the end of the stories. Story completion tasks rely not only on the logic of the story itself, but also requires implicit *commonsense knowledge* outside the story. To understand stories, human can use the information from both the story itself and other implicit sources such as commonsense knowledge and normative social behaviors. In this paper, we propose to imitate such behaviors to incorporate structured commonsense knowledge to aid the story ending prediction.

Recently, (Mostafazadeh et al. 2017) introduced a ROCStories dataset as a benchmark for evaluating models' ability to understand the narrative structures of a story, where the model is asked to select the correct ending from two candidates for a given story. To solve this task, both traditional machine learning approaches (Schwartz et al. 2017) and neural network models (Cai, Tu, and Gimpel 2017) have been used. Some works also exploit information such as sentiment and topic words (Chaturvedi et al. 2017) and event frames (Li et al. 2018). Recently, there has been work (Radford et al. 2018) that leverages large unlabeled corpus, like the BooksCorpus (Zhu et al. 2015) dataset, to improve the performance. However, none of them explicitly uses structured commonsense knowledge, which humans would naturally incorporate to improve model performance.



(a) An example story

(b) Clues in ConceptNet

Figure 1: (a) shows an example story from ROCStories dataset, words in colors are key-words. (b) shows the key-words and their relations in ConceptNet Knowledge Graph

Figure 1(a) shows a typical example in ROCStories dataset: a story about Dan and his parents. The blue words are key-words in the body of the story, and the red word is the key-word in the correct story ending. Figure 1(b) shows the (implicit) relations among these key-words, which are obtained as a subgraph from ConceptNet (Speer, Chin, and Havasi 2017), a commonsense knowledge base. By incorporating such structured external commonsense knowledge, we are able to discover strong associations between these

[*]This work was finished while Jiaao Chen was an intern in University of California, Davis.

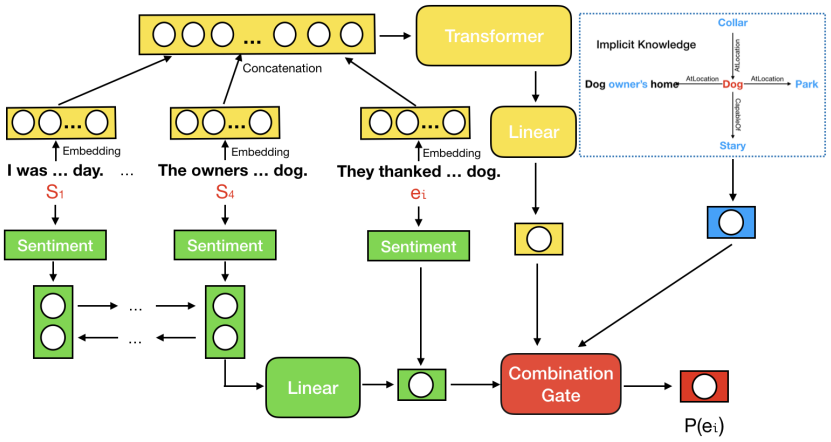


Figure 2: Our proposed model architecture. The inputs: S_1 through S_4 denote the story body, and e_i $(i = 1, 2)$ denotes two candidate endings. The bottom-left component encodes sentiment evolution information (green), the top-left component models the narrative sequence (yellow), and the top-right component integrates commonsense knowledge (blue). The combination gate in the bottom-right integrates all three types of information and outputs the probability on which ending is correct.

keywords and correctly predict the story ending. Note that these associations are not available from the story itself.

To solve the story completion task, we propose a neural network model that integrates three types of information: (i) narrative sequence, (ii) sentiment evolution, and (iii) commonsense knowledge. The clues in narrative chain are captured by a transformer decoder, constructed from a pretrained language model. The sentiment prediction is obtained by using a LSTM model. Additionally, the commonsense knowledge is extracted from an existing structured knowledge base, ConceptNet. In particular, we use a combination gate to integrate all the information and train the model in an end-to-end manner. Experiments demonstrate the improved performance of our model on the task.

Related Work

Our work on story completion is closely related to several research areas such as reading comprehension, sentiment analysis and commonsense knowledge integration, which will be briefly reviewed as below.

Reading Comprehension is the ability to process text, understand its meaning, and to integrate it with what the readers already know. It has been an important field in NLP for a long time. The SQuAD dataset (Rajpurkar, Jia, and Liang 2018) presents a task to locate the correct answer to a question in a context document and recognizes unanswerable questions. The RACE dataset (Lai et al. 2017), which is constructed from Chinese Students English Examination, introduces another task that requires not only retrieval but also reasoning. Usually they are solved by match-based model like QANET (Yu et al. 2018), hierarchical attention model like HAF (Zhu et al. 2018), and dynamic fusion based model like DFN (Xu et al. 2017). Also there exists more relevant research on story comprehension such as event understanding of narrative plots (Chambers and Jurafsky 2009) and character personas (Valls-Vargas, Ontañón, and Zhu 2015).

Sentiment Analysis aims to determine the attitude of a speaker (or a writer) with respect to some topic, the overall contextual polarity, or emotional reaction to a document, interaction or event. There have been rich studies on this field, such as learning word vectors for sentiment analysis (Maas et al. 2011) and recognizing contextual polarity in a phrase-level (Wilson, Wiebe, and Hoffmann 2005). Recently, researchers studied large-scale sentiment analysis across news and blogs (Godbole, Srinivasaiah, and Skiena 2007), and also studied opinion mining on twitter (Patodkar and Sheikh 2010). Additionally, there have been studies focused on joint learning for better performance, such as detecting sentiment and topic simultaneously from text (Lin and He 2009).

Commonsense Knowledge Integration If machines receive information from a commonsense knowledge base, they become more powerful for many tasks like reasoning (Bagherinezhad et al. 2016), dialogue generation (Liu et al. 2018) and cloze style reading comprehension (Mihaylov and Frank 2018). Related works include (Bagherinezhad et al. 2016), which builds a knowledge graph and uses it to deduce the size of objects (Bagherinezhad et al. 2016), in addiiton to (Zhu et al. 2017), in which a music knowledge graph is built for a single round dialogue system. There are several ways to incorporate external knowledge base (e.g., ConceptNet). For example, (Speer and Lowry-Duda 2017) uses a knowledge based word embedding, (Young et al. 2017) employs tri-LSTMs to encode the knowledge triple, and (Zhou et al. 2018) and (Mihaylov and Frank 2018) apply graph attention embedding to encode sub-graphs from a knowledge base. However, their work does not involve narrative completion.

Story Completion Traditional machine learning methods have been used to solve ROCStory Cloze Task such as (Schwartz et al. 2017). To improve the performance, features like topic words and sentiment score are also extracted and incorporated (Chaturvedi et al. 2017). Neural network models have also been applied to this task (e.g., (Huang et al.

2013) and (Cai, Tu, and Gimpel 2017)), which use LSTM to encode different parts of the story and calculate their similarities. In addition, (Li et al. 2018) introduces event frame to their model and leverages five different embeddings. Finally, (Radford et al. 2018) develops a transformer model and achieves state-of-the-art performance on ROCStories, where the transformer was pretrained on BooksCorpus (a large unlabeled corpus) and finetuned on ROCStories.

Proposed Model

For a given story $S = \{s_1, s_2, ..., s_L\}$ consisting of a sequence of L sentences, our task is to select the correct ending out of two candidates, e_1 and e_2, so that the completed story is reasonable and consistent. On the face of it, the problem can be understood as a standard binary classification problem. However, learning binary classifier with standard NLP techniques on the explicit information in the story is not sufficient. This is because correctly predicting the story ending usually requires reasoning with implicit commonsense knowledge. Therefore, we develop a neural network model to predict the story ending by integrating three sources of information: narrative sequence, sentiment evolution and structured commonsense knowledge (see Figure 2). Note that the first two types of information are explicit in the story while the third type is implicit and has to be imported from external source such as a knowledge base. In this section, we will explain how we exploit these three information sources and integrate them to make the final prediction.

Narrative Sequence

To describe a consistent story, plots should be planned in a logically reasonable sequence; that is there should be a narrative chain between different characters in the story. This is illustrated in the example in Figure 3, where words in red are events and words in blue are characters. The story chain, "*Agatha wanted pet birds* $\rightarrow$ *Agatha purchased pet finches* $\rightarrow$ *Agatha couldn't stand noise* $\rightarrow$ *mess was worse* $\rightarrow$ *Agatha return pet birds*", describes a more coherent and reasonable story than " *Agatha wanted pet birds* $\rightarrow$ *Agatha purchased pet finches* $\rightarrow$ *Agatha couldn't stand noise* $\rightarrow$ *mess was worse* $\rightarrow$ *Agatha buy two more*". When Agatha could not stand the noise, it is more likely for her to give these birds away rather than buy more. Therefore, developing a better semantic representation for narrative chains is important for us to predict the right endings.

Inspired by the recent research from OpenAI (Radford et al. 2018) on forming semantic representations of narrative sequences, we first pre-train a high-capacity language model on a large unlabeled corpus of text to learn the general information hidden in the context, and then fine-tune the model on this story completion task.

Given a large corpus of tokens $C = \{c_1, c_2, ..., c_n\}$, we can pre-train a language model to maximize the likelihood :

$$L_{lm}(C) = \sum_i \log P_l(c_i | c_{i-k}, ..., c_{i-1}; \theta) \quad (1)$$

where k is the window size, and the conditional probability P_l is modeled using a neural network with parameters θ.

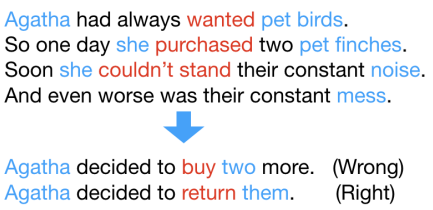


Figure 3: An example story in the ROCStories dataset

Similar to (Radford et al. 2018), we use a multi-layer transformer decoder with multi-headed self-attention for the language model:

$$h_0 = CW_e + W_p \quad (2)$$

$$h_l = transformer(h_{l-1}), l \in [1, M] \quad (3)$$

$$P(c) = softmax(h_M W_e^T) \quad (4)$$

where $C = \{c_1, c_2, ..., c_n\}$ are tokens in corpus, W_e is the token embedding matrix, W_p is the position embedding matrix and M is the number of transformer blocks.

We use the pre-trained parameters released by OpenAI [1] as the initialization for the transformer decoder. We adapt these parameters to our classification task. For each candidate story $(s_1, s_2, s_3, s_4, e_i)$ (i.e., the story body followed by one candidate ending), we serialize it into a sequence of tokens $X = \{x_1, ..., x_k\}$, where k is the number of tokens. Then the fine-tuned transformer takes X as its input and outputs the probability of e_i being the correct ending:

$$P_N(y | s_1, ..., s_4, e_i) = softmax(W_M h_M^k + b_M) \quad (5)$$

where $y \in \{0, 1\}$ is the label indicating whether e_i is the correct ending, h_M^k denotes the hidden representation at the M-th layer of the transformer associated with the k-th token, and W_M and b_M are parameters in the linear output layer.

Sentiment Evolution

Besides narrative sequence, getting a good sentiment prediction model is also important for choosing the correct endings. Note that stories are different from other objective texts (e.g., news), as they have emotions within the context. Usually there is a sentiment evolution when a storyline is being revealed (Vonnegut 1981) .

First, we pre-train a sentiment prediction model using the training set of the ROCStories, which does not have alternative endings (i.e., no negative samples). Given a five-sentence story $S = \{s_1, s_2, s_3, s_4, s_5\}$, we take the first four sentences as the body B and the last sentence as the ending e. We extract the sentiment polarity of each sentence by utilizing a lexicon and rule-based sentiment analysis tool (VADER) (Hutto and Gilbert 2014):

$$E_i = \text{VADER}(s_i), i \in [1, 5] \quad (6)$$

where E_i is a vector of three elements including probabilities of the i-th sentence being positive, negative and neutral.

[1] https://github.com/openai/finetune-transformer-lm

Then, we use a Long Short-Term Memory (LSTM) neural network to encode the sentence sentiments E_i with its context into the hidden state h_i, which summarizes the contextual sentiment information around the sentence s_i. And we use the last hidden state h_4 to predict the sentiment vector E_p in the ending e:

$$h_i = \text{LSTM}(E_i, h_{i-1}), i \in [1, 4] \quad (7)$$

$$E_p = softmax(W_e h_4 + b_e) \quad (8)$$

We train the sentiment model by maximizing the cosine similarity between the predicted sentiment vector E_p and the sentiment vector E_5 of the correct ending:

$$sim(S) = \frac{E_p \cdot E_5}{\|E_p\|_2 \cdot \|E_5\|_2} \quad (9)$$

Afterwards, we adapt the parameters to the story ending selection task and calculate the following conditional probability P_S:

$$P_S(y|s_1, ..., s_4, e_i) = softmax(E_p W_s E_e) \quad (10)$$

where $S = \{s_1, s_2, s_3, s_4\}$ is the body, e_i is the candidate ending, E_p is the predicted sentiment vector, E_e is the sentiment vector extracted from ending e_i, and W_s is the similarity matrix to be learned.

Commonsense Knowledge

Narrative sequence and sentiment evolution, though useful, are not sufficient to make correct predictions. In a typical story, newly introduced key-words may not be explained in the story because story-writers are not given enough narrative space and time to develop and describe them (Martin and George 2000). In fact, there are many hidden relationships among key-words in natural stories. In Figure 1 (a), although the key-word "diet" in the ending is not mentioned in the body, there are hidden relationships among "diet", "overweight" and "unhealthy" as shown in Figure 1 (b). When this kind of implicit information is uncovered in the model, it is easier to predict the correct story ending.

We leverage the implicit knowledge by using a numberbatch word embedding (Speer, Chin, and Havasi 2017), which is trained on data from ConceptNet, word2vec, GloVe, and OpenSubtitles. The numberbatch achieves good performance on tasks related to commonsense knowledge (Speer and Lowry-Duda 2017). For instance, the cosine similarity between "diet" and "overweight" in numberbatch is 0.453, but it is 0.326 in GloVe. This is because numberbatch makes use of the relationship between them as shown in Figure 1 (b) while GloVe does not.

Given the body $S = \{s_1, s_2, s_3, s_4\}$, a candidate ending e_i and the label y, we tokenize each sentence using NLTK and Standford's CoreNLP tools (Manning et al. 2014). After deleting the stop words, we calculate the knowledge distance vector D between the candidate ending and the body by Algorithm 1. We compute the similarity between two key-words using the cosine similarity of their vector space representations in numberbatch. For each sentence s_i in the body, we then quantify the distance with the ending using averaged alignment score of every key-word in the ending.

Algorithm 1 Knowledge distance computation

for all sentence s_j such that $s_j \in S$ **do**
  $distance_j = 0$
  $num = 0$
  for all word w such that $w \in e_i$ **do**
    $max_d = 0$
    $num += 1$
    for all word u such that $u \in s_j$ **do**
      if $stem(w) != stem(u)$ **then**
        d = cosine similarity(w, u)
        if $d > max_d$ **then** $max_d = d$
        end if
      end if
    end for
    $distance_j += max_d$
  end for
  $distance_j /= num$
end for
return $(distance_1, ..., distance_4)$

Then we use a linear layer to model the conditional probability P_C:

$$P_C(y|s_1, ..., s_4, e_i) = softmax(W_d D + b_d) \quad (11)$$

where W_d and b_d are parameters in the linear output layer, and D is the four-dimensional distance vector.

Combination Gate

Finally, we predict the story ending by combining the above three sources of information. We utilize the feature vectors h_M^k in the narrative sequence, E_e in the sentiment evolution, and D in the commonsense knowledge and calculate their cosine similarities. Then we concatenate them into a vector g. We use a linear layer to model the combination gate and use that gate to combine three conditional probabilities.

$$G = softmax(W_g g + b_g) \quad (12)$$

$$\tilde{P}(y|s_1, ..., s_4, e_i) = softmax(sum(G \odot [P_N; P_S; P_C])) \quad (13)$$

where W_g and b_g are parameters in the linear layer, (P_N, P_S, P_C) are the three probabilities modeled in (5), (10) and (11), G is the hidden variable that weighs three different conditional probabilities and $\odot$ is element-wise multiplication.

Finally, since each of the three components (P_N, P_S and P_C) are either pre-trained on a separate corpus or individually tuned on the task, we fine-tune the entire model in an end-to-end manner by minimizing the following cost:

$$\tilde{L} = L_{cm}(S) - \lambda * L_{lm}(C) \quad (14)$$

where $L_{cm}(s) = \sum -ylog(\tilde{P})$ is the cross-entropy between the final predicted probability and the true label, L_{lm} is a regularization term of language model cost, and λ is the regularization parameter.

Sentence	Number of words	Number of keywords
s_1	8.9	6.2
s_2	9.9	6.5
s_3	10.2	6.7
s_4	10.0	6.5
e_1	10.5	5.7
e_2	10.3	5.8

Table 1: The average number of words and key-words exist in ConceptNet in each sentence of the story

Dataset

We evaluated our model on ROCStories (Mostafazadeh et al. 2017), a publicly available collection of commonsense short stories. This corpus consists of 100,000 five-sentence stories. Each story logically follows everyday topics created by Amazon Mechanical Turk (MTurk) workers. These stories contain a variety of commonsense causal and temporal relations between everyday events. Writers also develop an additional 3,742 stories which contain a four-sentence-long body and two candidate endings. The endings were collected by asking MTurk workers to write both a right ending and a wrong ending after eliminating original endings of given short stories. Both endings were required to include at least one character from the main story line and to make logical sense. and were tested on AMT to ensure the quality. The published ROCStories dataset [2] is constructed with ROCStories as a training set that includes 98,162 stories that exclude candidate wrong endings, an evaluation set, and a test set, which have the same structure (1 body + 2 candidate endings) and a size of 1,871.

We find that the dataset contains 43,095 unique words, and 28,012 key-words in ConceptNet. The average number of words and key-words in ConceptNet for each sentence are shown in Table 1. s_1, s_2, s_3 and s_4 are four sentences in the body of stories. e_1 and e_2 are the two candidate endings. A large portion (65%) of words mentioned in stories are key-words in ConceptNet. Thus we believe ConceptNet can provide additional information to the model.

In our experiments, we use a training set which does not have candidate endings to pre-train the sentiment prediction model. For learning to select the right ending, we randomly split 80% of stories with two candidates endings in ROCStories evaluation set as our training set (1,479 cases), 20% of stories in ROCStories evaluation set as our validation set (374 cases). And we utilize the ROCStories test set as our testing set (1,871 cases).

Experiments

Baselines

We use the following models as our baselines:

Msap(Schwartz et al. 2017): Msap uses a linear classifier based on language modeling probabilities of the entire story, and utilizes linguistic features of the ending sentences. These ending "style" features include sentence length, word and character n-gram in each candidate ending (independent of story).

HCM(Chaturvedi et al. 2017): HCM uses FC-SemLM (Peng and Roth 2016) in order to represent events in the story, learns sentiment trajectories in a form of N-gram language model, and uses topic-words' GloVe to extract topical consistency feature. It uses Expectation-Maximization for training.

DSSM(Huang et al. 2013): DSSM first uses two deep neural networks to project the context and the candidate endings into the same vector space, and ending choices based on the cosine similarity of the context.

Cai(Cai, Tu, and Gimpel 2017): Cai uses BiLSTM RNN with attention mechanisms to encode the body and ending of the story separately and uses a cosine similarity between their representations to calculate the score for each ending during selection process.

SeqMANN(Li et al. 2018): SeqMANN uses a multi-attention neural network and introduces semantic sequence information extracted from FC-SemLM as external knowledge. The embedding layer concatenates five representations including word embedding, character feature, part-of-speech (POS) tagging, sentiment polarity and negation. The model uses DenseNet to match body with an ending.

FTLM(Radford et al. 2018): FTLM solves the stories cloze test by pre-training a language model using a multi-layer transformer on a diverse corpus of unlabeled text, followed by discriminative fine-tuning.

[2] http://cs.rochester.edu/nlp/rocstories

Model	Accuracy(%)
Msap(Schwartz et al. 2017)	75.2
HCM (Chaturvedi et al. 2017)	77.6
DSSM (Huang et al. 2013)	58.5
Cai (Cai, Tu, and Gimpel 2017)	74.7
SeqMANN (Li et al. 2018)	84.7
FTLM (Radford et al. 2018)	86.5
Our Model(Plot&End)	78.4
Our Model(Full Story)	**87.6***

Table 2: Performance comparison with baselines, *indicates that the model is significantly better than best baseline model

Experimental Settings

We tune the hyper parameters of models on the validation set. Specifically, we set the dimension of LSTM for sentiment prediction to 64. We use a mini-batch size of 8, and Adam to train all parameters. The learning rate is set to 0.001 initially with a decay rate of 0.5 per epoch.

Results

We evaluated baselines and our model using accuracy as the metric on the ROCStories dataset, and summarized these results in Table 2. The linear classifier with language model, **Msap**, achieved an accuracy of 75.2%. When adding additional features, such as sentiment trajectories and topic words to traditional machine learning methods, **HCM** achieved an accuracy of 77.6%. Recently, more neural

Types of information	Accuracy(%)
Narrative	85.3
Sentiment	58.7
Knowledge	63.8
Our Model(All Types)	**87.6**

Table 3: Performance on only using one type of information

Types of information	Accuracy(%)
Our Model(All Types)	**87.6**
- Narrative	65.9
- Sentiment	87.2
- Knowledge	85.6

Table 4: Performance on stripping one type of information, e.g. "- Sentiment" means removing sentiment information.

network-based models are used. **DSSM** simply used a deep structured semantic model to learn representations for both bodies and endings only achieved an accuracy of 58.5%. Utilizing **Cai** improved neural model performance to 74.7% by applying attention mechanisms on a BiLSTM RNN structure. **SeqMANN** further improved the performance to 84.7%, when combining more information from embedding layers, like character features, part-of-speech (POS) tagging features, sentiment polarity, negation information and some external knowledge of semantic sequence. Researchers also improved model performance by pre-training word embeddings on external large corpus. **FTLM** pre-trained a language model on a large unlabeled corpus and fine-tuned on the ROCStories dataset, and achieved an accuracy of 86.5%.

We tried two different ways to construct narrative sequence features: Plot&End and FullStory. Plot&End encodes the body and ending of a story separately and then computes their cosine similarity. We use a hierarchy structure to encode the four body sentences. However using such encoding method, our model only achieved an accuracy of 78.4%. One possible reason is that the relation between sentences learned through pre-trained language models are not fully explored if we encode each sentence separately. FullStory encodes all five sentences together. Our model achieved the best performance when using FullStory mode to encode narrative sequence information. We achieved an accuracy of 87.6%, outperforming all baseline models. Such improvement may come from the full use of the pre-trained transformer block, as well as the incorporation of the structured commonsense knowledge and sentiment information in the model.

Ablation Study

We conducted another two groups of experiments to investigate the contribution of the three different types of information: narrative sequence, sentiment evolution and commonsense knowledge. First, we measure the accuracy of only using one type of information at a time and describe the result in Table 3. When we use just one type of information, the performances are worse than when using all of the information, suggesting a single type of information is insufficient for story ending selection. We also measure the performance of our model by stripping one type of information at a time and display the results in Table 4. We observe that by removing the narrative sequence information, the model performance decreases most significantly. We suspect this is because the narrative chain is the key element that differentiates a story from other types of writing. Therefore, removing narrative sequence information makes it difficult to predict the story ending. If we only use the narrative sequence information, the performance is 85.3%. When commonsense knowledge is added to the model on top of the narrative sequence information, the performance improves to 87.2% which is statistically significant. When sentiment evolution information is added, the model only improves to 87.6%. We speculate this is because the pre-trained language model from narrative sequence information may already capture some sentiment information, as it is trained on an ensemble of several large corpus. This suggests that commonsense knowledge has a large impact on narrative prediction task.

Case Study

We present several examples to describe the decision made at the combination gate. All the examples are shown in Table 5.

The first story shows how narrative sequence can be the key in detecting the coherent story ending. This one tells a story of Agatha and birds. As we have analyzed in the narrative sequence, the narrative chain is apparently the most effective clue in deciding the right ending. In the combination gate, the narrative part's weight is 0.5135, which is larger than the sentiment component's weight, 0.2214 as well as the commonsense component's weight of 0.2633. The conditional probability of the correct ending given the narrative information is 0.8634, which is much larger than the wrong ending. As both sentences' sentiments are neutral, the sentiment information is not useful . And as the word "buy" has closer relation to "want" and "purchase" mentioned in the sentence body than the word,"return", the commonsense knowledge actually makes the wrong decision which gives slightly higher probabilities to the wrong ending(0.5642).

The second story shows why and how sentiment evolution is influencing the final performance. It is a story about Jackson's beard: Jackson wanted to grow a beard regardless of what his friends said, and he was satisfied with his bushy, thick beard. Clearly the emotions between the two candidate endings are different. Based on the rule of consistent sentiment evolution, an appropriate ending should have a positive emotion rather than a negative emotion. The output of our model shows that in the combination gate, the sentiment evolution component received the largest weight, 0.4880, while the narrative sequence and the commonsense knowledge component have a weight of 0.2287 and 0.2833. Finally, the probability of the correct ending is 0.5360, larger than that of the wrong ending which is 0.4640 in sentiment part. Whereas in the narrative sequence component, the probability of the correct option is 0.4640, smaller than the wrong ending which is 0.5360. Other models like FTLM that only rely on narrative sequence will make the wrong decision in this case. The probabilities of the commonsense knowledge

Body	Correct ending	Wrong ending
Agatha had always wanted pet birds. So one day she purchased two pet finches. Soon she couldn't stand their constant noise. And even worse was their constant mess.	Agatha decided to return them.	Agatha decided to buy two more.
Jackson had always wanted to grow a beard. His friends told him that a beard would look bad, but he ignored them. Jackson didn't shave for a month and he grew a bushy, thick beard. Admiring himself in the mirror, Jackson felt satisfied.	He was glad that he hadn't listened to his friends.	He was ashamed of himself.
I was walking through Central Park on a fall day. I found a stray dog with a collar. I called the number on the collar and talked to the owners. The owners came to the park to pick up their dog.	They thanked me very much for finding their dog.	They let me keep it.

Table 5: Three examples from ROCStories. The first column is the body of the story, the second column is the correct ending, and the third column is the wrong ending.

component is 0.5257 versus 0.4725. Through combination gate, our model mainly relies on the sentiment to make a selection. As a result, it will identify the right ending despite other components influence toward a wrong decision.

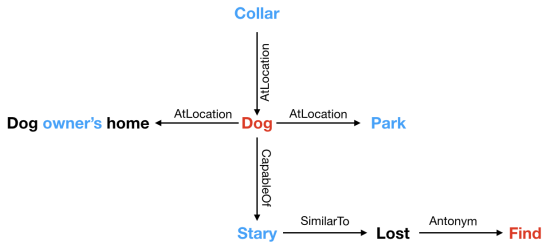


Figure 4: Sub-graph in ConceptNet

The third example presents the roles commonsense knowledge plays in our model. It tells a story about a person finding a dog. The sentiments of the two candidates are both neutral again. But based on the knowledge graph in ConceptNet, shown in Figure 4, there exists many relations between the correct ending and the story body. The key-words in the ending are in red, and the key-words in the story body are in blue. The key-words such as "stray" and "collar" are highly associated with "dog" and "find" in the correct ending. The result shows that the gate gives the commonsense knowledge component a weight of 0.5156, which is the largest among the three components. The conditional probability of the correct ending considering commonsense information (0.5540) is larger than the wrong ending as we expected. In this case, the narrative sequence component makes the wrong decision, which gives higher probabilities to the wrong ending (0.5283). Thus models like FTLM which only consider narrative chain will identify the wrong ending. However, as the combination gate learns to trust the commonsense knowledge component in this example more, our model still predicts the correct ending.

We can see that our model is able to learn to rely on different information types based on the content of different stories. We obtain such model effectiveness by using a combination gate to fuse all three types of information, and in doing so, understand how all three are imperative in covering all possible variations in the dataset.

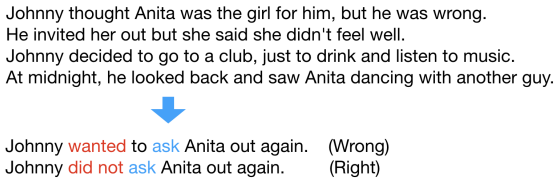


Figure 5: An example involves negation in ROCStories

However, it is still challenging for our model to handle the stories that have negations. Figure 5 shows an example. It tells a story between Johnny and Anita. But the only difference between two candidate endings is the negation word. Even when fusing three types of information, our model still cannot get the answer right. Because both event chains are about "asking Anita out", they are both neutral in sentiment, and the key-words in these two endings are the same as well. In the future, we plan to incorporate natural language inference information to the model to handle such cases.

Conclusion

Narrative completion is a complex task that requires both explicit and implicit knowledge. We proposed a neural network model that utilized a combination gate to fuse three types of information including: narrative sequence, sentiment evolution and structured commonsense knowledge to predict story endings. The model outperformed state-of-the-art methods. We found that introducing external knowledge such as structured commonsense knowledge helps narrative completion.

Acknowledgments

This work was funded by CCF-Tencent Rhino Bird Fund. We would like to thank anonymous reviewers and those who helped improve the draft.

References

Bagherinezhad, H.; Hajishirzi, H.; Choi, Y.; and Farhadi, A. 2016. Are elephants bigger than butterflies? reasoning about sizes of objects. *CoRR* abs/1602.00753.

Cai, Z.; Tu, L.; and Gimpel, K. 2017. Pay attention to the ending:strong neural baselines for the roc story cloze task. In *ACL*.

Chambers, N., and Jurafsky, D. 2009. Unsupervised learning of narrative schemas and their participants. In *ACL*.

Chaturvedi, S.; Peng, H.; Roth, D.; and Roth, D. 2017. Story comprehension for predicting what happens next. In *EMNLP*.

Godbole, N.; Srinivasaiah, M.; and Skiena, S. 2007. Large-scale sentiment analysis for news and blogs. In *ICWSM*.

Huang, P.-S.; He, X.; Gao, J.; Deng, L.; Acero, A.; and Heck, L. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, 2333–2338. New York, NY, USA: ACM.

Hutto, C. J., and Gilbert, E. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *ICWSM*.

Jockers, M. L. 2013. *Macroanalysis: Digital Methods and Literary History*. Champaign, IL, USA: University of Illinois Press, 1st edition.

Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; and Hovy, E. 2017. Race: Large-scale reading comprehension dataset from examinations. In *EMNLP*.

Li, Q.; Li, Z.; Wei, J.-M.; Gu, Y.; Jatowt, A.; and Yang, Z. 2018. A multi-attention based neural network with external knowledge for story ending predicting task. In *ICCL*.

Lin, C., and He, Y. 2009. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM.

Liu, Q.; Feng, Y.; Chen, H.; Ren, Z.; Yin, D.; and Liu, S. 2018. Knowledge diffusion for neural dialogue generation. In *ACL*.

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 *ACL*.

Manning, C. D.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S. J.; and McClosky, D. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL System Demonstrations*, 55–60.

Martin, W; B., and George, G. 2000. *Qualitative Researching with Text, Image and Sound: A Practical Handbook*. SAGE Publication.

Mihaylov, T., and Frank, A. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *ACL 2018*, 821–832.

Mostafazadeh, N.; Roth, M.; Louis, A.; Chambers, N.; and Allen, J. 2017. Lsdsem 2017 shared task: The story cloze test. In *ACL*, 46–51.

Patodkar, V. N., and Sheikh, I. R. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *LREC*.

Peng, H., and Roth, D. 2016. Two discourse driven language models for semantics. In *ACL*.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training. In *Arxiv*.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know what you don't know: Unanswerable questions for squad. In *ACL*.

Schubert, L. K., and Hwang, C. H. 2000. Episodic logic meets little red riding hood: A comprehensive natural representation for language understanding. In Iwańska, L. M., and Shapiro, S. C., eds., *Natural language processing and knowledge representation*. Cambridge, MA, USA: MIT Press. 111–174.

Schwartz, R.; Sap, M.; Konstas, I.; Zilles, L.; Choi, Y.; and Smith, N. A. 2017. Story cloze task: Uw nlp system. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*.

Speer, R., and Lowry-Duda, J. 2017. Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. *CoRR* abs/1704.03560.

Speer, R.; Chin, J.; and Havasi, C. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, 4444–4451.

Valls-Vargas, J.; Ontañón, S.; and Zhu, J. 2015. Narrative hermeneutic circle: Improving character role identification from natural language text via feedback loops. In *IJCAI*.

Vonnegut, K. 1981. *Palm Sunday*. RosettaBooks LLC.

Wilson, T.; Wiebe, J.; and Hoffmann, P. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT*.

Xu, Y.; Liu, J.; Gao, J.; Shen, Y.; and Liu, X. 2017. Towards human-level machine reading comprehension: Reasoning and inference with multiple strategies. *CoRR* abs/1711.04964.

Young, T.; Cambria, E.; Chaturvedi, I.; Huang, M.; Zhou, H.; and Biswas, S. 2017. Augmenting end-to-end dialog systems with commonsense knowledge. *CoRR* abs/1709.05453. Withdrawn.

Yu, A. W.; Dohan, D.; Luong, M.; Zhao, R.; Chen, K.; Norouzi, M.; and Le, Q. V. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *CoRR* abs/1804.09541.

Zhou, H.; Young, T.; Huang, M.; Zhao, H.; Xu, J.; and Zhu, X. 2018. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI-18*, 4623–4629.

Zhu, Y.; Kiros, R.; Zemel, R. S.; Salakhutdinov, R.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *CoRR* abs/1506.06724.

Zhu, W.; Mo, K.; Zhang, Y.; Zhu, Z.; Peng, X.; and Yang, Q. 2017. Flexible end-to-end dialogue system for knowledge grounded conversation. *CoRR* abs/1709.04264.

Zhu, H.; Wei, F.; Qin, B.; and Liu, T. 2018. Hierarchical attention flow for multiple-choice reading comprehension. In *AAAI*.