Story Ending Selection by Finding Hints from Pairwise Candidate Endings

Mantong Zhou,Minlie Huang,Xiaoyan Zhu

***Abstract*—The ability of story comprehension is a strong indicator of natural language understanding. Recently, Story Cloze Test has been introduced as a new task of machine reading comprehension, i.e., selecting a correct ending from two candidate endings given a four-sentence story context. Most existing methods for Story Cloze Test are essentially matching-based that operate by comparing an individual ending with a given context, therefore suffering from the evidence bias issue: both candidate endings can obtain supporting evidence from the story context, which misleads the classifier to choose an incorrect ending. To address this issue, we present a novel idea to improve story comprehension by utilizing the hints which are obtained through comparing two candidate endings. The proposed model firstly anticipates a feature vector for a possible ending solely based on the context, and then refines the feature prediction using the hints which encode the difference between two candidates. The candidate ending whose feature vector is more similar to the predicted ending vector is regarded as correct. Experimental results demonstrate that our approach can alleviate the evidence bias issue and improve story comprehension.**

***Index Terms*—Machine Reading Comprehension, Story Comprehension, Commonsense Reasoning, Neural Networks**

I. Introduction

STORY comprehension is a fundamental but challenging task in natural language understanding [1], which can enable computers to learn about social norms, human behavior, and commonsense knowledge. Recently, **Story Cloze Test (SCT)** [2], [3] was introduced to evaluate the machinery ability of story understanding, story generation, and script learning. Story comprehension differs significantly from previous machine comprehension tasks [4]–[7] in that SCT focuses on reasoning with implicit knowledge by requiring a system to select a correct ending from two candidates, given a four-sentence story context.

Most existing models [8], [9] for story ending prediction are motivated by lexical or semantic matching, through either attention mechanism or feature engineering, which search for *important* linkages between a story context and a candidate ending. They suffer from the issue of **evidence bias**: both the wrong and correct endings can obtain sufficient support from the story context. As illustrated in Fig. 1, the wrong ending (in red) and the correct ending (in green) can be supported by the red-colored evidence and the green-colored evidence in the story context, respectively. Thus, it is difficult for matching-based models to distinguish such cases. The situation is not rare because both correct and wrong endings are written to fit the world of a story in SCT during the construction of this corpus. For instance, the protagonist typically appears in both endings [3], [8], and the event in a wrong ending is also closely related to the story context. Hence, it is challenging to identify the correct ending since the wrong ending is also highly plausible.

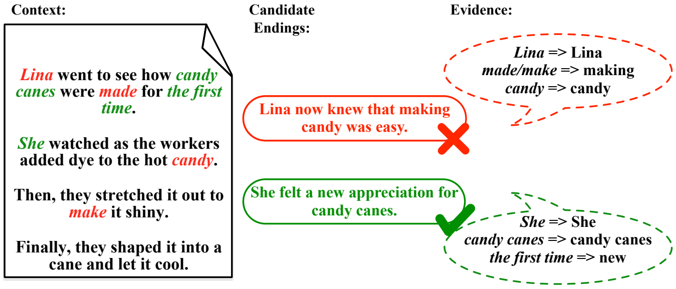
Context:
Lina went to see how candy canes were made for the first time.
She watched as the workers added dye to the hot candy.
Then, they stretched it out to make it shiny.
Finally, they shaped it into a cane and let it cool.
Candidate Endings:
Lina now knew that making candy was easy.
She felt a new appreciation for candy canes.
Evidence:
Lina => Lina
made/make => making
candy => candy
She => She
candy canes => candy canes
the first time => new

Fig. 1. **Evidence bias issue**: both a wrong ending (in red) and a correct ending (in green) can obtain sufficient evidence from the story context.

Statistics on 3,742 SCT stories show that, 35.9% words in correct endings can be found in the corresponding story contexts, but 36.2% words in wrong endings also appear in the contexts. In 34.9% stories, word overlap between the wrong ending and the context is larger than that between the correct ending and the context. Furthermore, we investigate whether terms (such as *'have lunch'*) in an ending are relevant to the context. Terms in an ending which have ConceptNet[1] relations to the terms in a story context are defined as related terms. In 37.7% stories, the wrong ending possesses more related terms than the correct ending. These simple statistics further demonstrates that the evidence bias issue is commonly observed in this corpus.

To avoid the evidence bias issue caused by individual matching, there is a natural two-stage process: first read the context solely and then consider the two candidates simultaneously. Inspired by FESLM [10] which used a language model to evaluate story endings, we argue that a story comprehension model should have the ability to anticipate a reasonable ending after reading the context. The ending selection thus depends on predicting a possible ending rather than merely matching an individual candidate ending to the context. However, anticipating proper endings solely based on the story context is difficult since there are too many basic elements describing

Mantong Zhou, Minlie Huang and Xiaoyan Zhu are with the Institute for Artificial Intelligence, State Key Lab of Intelligent Technology and Systems, Beijing National Research Center for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China, e-mail: (zmt.keke@gmail.com, aihuang@tsinghua.edu.cn, zxy-dcs@tsinghua.edu.cn).

Corresponding author: Minlie Huang.

[1] A commonsense knowledge base, see http://conceptnet.io.

the characters, events, and sentiments in a story. It is hard to decide which elements are useful to compose correct endings. Instead, if we have some **hints** about what is correct or reasonable, it would be easier to identify the correct ending. As shown in Fig. 1, since both candidates are accordant with the context with respect to the character, we should ignore the useless element (*'Lina / She'*→*'Lina / She'*) and emphasize the valuable ones attached to the correct candidate (*'first time'*→*'new appreciation'*). Accordingly, we can obtain hints by comparing two candidate endings and thus predict a correct ending more easily.

In this paper, we present a novel model for choosing a correct story ending from two candidates given the story context. Our model is a two-stage process: **First**, read the story context without any biased information and anticipate an ending; **Second**, refine the story ending prediction by utilizing hints via comparing two candidate endings. Our model makes a preliminary prediction of a reasonable ending and then refines the prediction using the hints that manifest the difference between two candidate endings. The contributions of this work are in three folds:

- We study the evidence bias issue which occurs in most existing methods for story ending selection. Our model considers both candidates simultaneously for story ending selection, which is the major departure from existing methods that compare a candidate ending separately with the story context.
- We propose an approach, HintNet, to improve story comprehension by employing story hints which explicitly encode the difference between two candidate endings. Different hint representations are studied, and the connections between hint representations and surface words are discovered.
- Experimental results show that HintNet alleviates the issue of evidence bias and improves story ending selection.

II. Related Works

Story comprehension is a challenging task in natural language understanding since narrative stories capture rich causal and temporal commonsense relations. Understanding stories involves textual semantic understanding, logical reasoning and natural text generation. A large body of work in story comprehension has focused on scripts learning [11]. Narrative chains [12], narrative schemas [13], script sequences [14], and relgrams [15] are proposed to learning narrative/event representations. Several groups have directly addressed script learning by focusing exclusively on the narrative cloze test [12], in which a system predicts a held-out event (a verb and its arguments) given a set of observed events. Previous works [16], [17] showed that language-modeling techniques perform well on original narrative cloze test. **Story Cloze Test (SCT)** [2], [3] is thus introduced as a new evaluation framework. Instead of predicting an event, the system is required to select a reasonable ending from two given candidates.

As a unique branch of machine reading comprehension, SCT requires to select a correct ending from two candidates given a story context. It is different from classic reading comprehension tasks in a form of question answering [4]–[6], which require a system to find the answer to a specific question in a given document. In classic reading comprehension tasks, most answers can be identified by supporting facts through matching a token span of the document to the question, without misleading evidence. By contrast, in SCT, the two candidate endings are both highly plausible, and a wrong ending can also obtain sufficient evidence from the story context, thus misleading matching-based classifiers.

Previous studies on Story Cloze Test can be roughly categorized into two lines: feature-based methods and neural models. Traditional feature-based methods for story understanding [18]–[21] adopted some shallow semantic features, such as n-grams and POS tags and trained a linear regression model to determine whether a candidate ending is plausible. Two recent works [9], [22] enhanced feature-based story understanding by modelling the coherence of a complete story from three perspectives, namely event, sentiment, and topic. MKR [9] exploited heterogeneous resources and computed the cost of inferring word spans in an ending from the corresponding matching spans in the context with some predefined rules. HCM [22] designed linguistic features over multiple semantic aspects to select correct endings. They used SemLM [23], N-gram model and topic-words' information to capture and measure event-sequence, sentiment trajectories and topical consistency, respectively. Although HCM overcame the shortcomings of simple feature-based classifiers via generative language models, these methods did not explicitly address the evidence bias issue.

Neural models for SCT represent a candidate ending and the context with low-dimensional dense vectors [24], and story ending selection is made by computing the vectors' similarity [8], [25] or by a binary classifier [24], [26]. To capture the deep meaning of highly recapitulative sentences better, attention based RNNs were proposed to represent the context and a candidate ending [8], [25]. HBiLSTM [8] built a hierarchical bidirectional LSTM model using attention mechanism to modify context representations according to the input candidate ending. Recent works [27], [28] pursued the same strategy as HBiLSTM but enriched synthesis word embeddings and applied more complicated attention mechanism. Recent state-of-the-art methods, including SeqMANN [28], FTLM (also known as OpenAI GPT) [29], and Commonsense-Model [30], showed that training on very large data can make remarkable improvements for Story Cloze Test. SeqMANN utilized external information to improve its basic multi-attention network. FTLM pre-trained a language model on a corpus of 74M sentences, then fine-tuned an additional discriminator on the story corpus. Commonsense-Model supplemented FTLM with commonsense knowledge and sentiment polarity to achieve better results. Although additional resources (either large language corpus or knowledge bases) have demonstrated substantial improvements on story ending prediction, there is still room to improve existing attention-based neural models with the given corpus alone. Though attention mechanism is effective to highlight supporting evidence in the context, it suffers from the evidence bias issue because a wrong ending can also match word spans in the story context.

Aakanksha [31] proposed stress tests for natural language inference (NLI or RTE) task to evaluate whether a model is robust to distractions in the form of adversarial examples. They found that lexical similarity is the major factor for the success of existing state-of-the-art models in natural language inference, which makes these models unreliable to the cases that are beyond lexical overlap. Their findings also inspire us to design a story comprehension model that can avoid biased lexical matching.

There are a few works on story generation, varying from phrase-based plot generation [32], [33], story ending generation [10], [34], [35], to entire story generation [36]. Rule-based methods [37], [38] and neural models with Sequence-to-Sequence framework [39], [40] are widely used. Though ending selection is different from ending generation, the findings in this paper may inspire other generation models to make use of the hint information that are important to make a reasonable and coherent story.

III. Model

A. Task Formulation and Model Overview

Task Formulation

Our task is formulated as follows: Given a four-sentence story context $\{S^1, S^2, S^3, S^4\}$, and two candidate endings S^1_{end} and S^2_{end}, the goal is to identify which ending is correct.

We approach the task as a **feature vector** prediction problem [24]. Each sentence S^i, including the candidate endings is represented by a vector $\boldsymbol{s}^i \in \mathbb{R}^{d_s}$, and the model aims to predict a feature vector $\boldsymbol{s}_p \in \mathbb{R}^{d_s}$ that represents a possible ending. The final decision is made by choosing a candidate ending whose sentence vector is closer to the predicted feature vector $\boldsymbol{s}_p$.

The benefit to predict an ending feature vector instead of performing a two-class classification lies in that we can utilize the feature vector in other succeeding tasks such as story ending generation.

Model Overview

Our model, named as **HintNet**, incorporates the hints from two competing candidates to achieve a better understanding of the story. As illustrated in Fig. 2, HintNet applies a two-stage procedure. At the first stage, the model makes a **preliminary prediction**, which generates a feature vector $\boldsymbol{s}_{pp}$ solely based on the story context without any information from the candidate endings. The predicted vector $\boldsymbol{s}_{pp}$ can be viewed as a feature representation that may imply a correct ending merely from the story context. At the second stage, a latent variable will be used to encode the hint to obtain a **refined prediction**, $\boldsymbol{s}_{rp}$, which improves the preliminary prediction to make the predicted vector closer to the gold ending. The hint information encodes the difference between two candidate endings to avoid the *evidence bias issue*, with a purpose of supporting the correct ending, meanwhile inhibiting other misleading information in the wrong ending.

The two-stage procedure is inspired by the story comprehension process mostly happened in humans: firstly, making a preliminary scan of the story context and anticipating a possible ending, and secondly, obtaining some hints from candidate endings to gain better story comprehension. In this manner, it is easier for us to decide which ending is more reasonable and coherent to the story context.

B. Preliminary Prediction

At the first stage, HintNet encodes each sentence in a story context into a vector and then predicts a feature vector to imply a possible story ending.

Sentence Encoder

HintNet adopts a bidirectional LSTM [41], named as *Sentence Encoder*, to encode a sentence to a vector representation $\boldsymbol{s} \in \mathbb{R}^{d_s}$. To emphasize informative keywords in a long sentence, a self-attention mechanism is applied.

Specifically, for sentence S^i, we concatenate the hidden states of the forward and the backward LSTMs at each position to obtain the corresponding representation as $\boldsymbol{v}^i_j$:

$$\boldsymbol{f}^i_j = \mathbf{LSTM}^f(\boldsymbol{f}^i_{j-1}, \boldsymbol{e}(x^i_j)) \tag{1}$$

$$\boldsymbol{b}^i_j = \mathbf{LSTM}^b(\boldsymbol{b}^i_{j+1}, \boldsymbol{e}(x^i_j)) \tag{2}$$

$$\boldsymbol{v}^i_j = [\boldsymbol{f}^i_j; \boldsymbol{b}^i_j] \tag{3}$$

where $\boldsymbol{e}(x^i_j)$ denotes the word embedding of word x^i_j, and $[\boldsymbol{a}; \boldsymbol{b}]$ stands for the concatenation of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

We then concatenate the last hidden states $\boldsymbol{f}^i_{|S^i|}$ and $\boldsymbol{b}^i_1$ of the two LSTMs to form an intermediate sentence vector $\boldsymbol{u}^i$ (same as the encoding of a query u in [5]). The sentence vector $\boldsymbol{s}^i$ is thereby constructed on top of $\boldsymbol{u}^i$ as follows:

$$\boldsymbol{u}^i = [\boldsymbol{f}^i_{|S^i|}; \boldsymbol{b}^i_1] \tag{4}$$

$$a^i_j = (\boldsymbol{v}^i_j)^T \boldsymbol{M} \boldsymbol{u}^i \tag{5}$$

$$\boldsymbol{s}^i = \sum_j^{|S^i|} \frac{exp(a^i_j)}{\sum_k exp(a^i_k)} \boldsymbol{v}^i_j \tag{6}$$

Preliminary Ending Predictor

This ending predictor obtains a vector $\boldsymbol{s}_{story} \in \mathbb{R}^{d_s}$ to represent the story context and predicts a feature vector $\boldsymbol{s}_{pp}$ which implies a possible story ending.

The four-sentence story context is encoded by another BiLSTM (*Story Encoder*). To fully leverage the hint vector subsequently, this module firstly transforms $\boldsymbol{s}_{story}$ into a latent variable $\boldsymbol{z}$, and then derives a feature vector $\boldsymbol{s}_{pp}$ as the preliminary prediction, as follows:

$$\boldsymbol{s}_{story} = \mathbf{BiLSTM}(\boldsymbol{s}^1, \boldsymbol{s}^2, \boldsymbol{s}^3, \boldsymbol{s}^4) \tag{7}$$

$$\boldsymbol{z} = \mathbf{F}(\boldsymbol{s}_{story}) \tag{8}$$

$$\boldsymbol{s}_{pp} = \mathbf{G}(\boldsymbol{z}) \tag{9}$$

where we apply an encoding network $\mathbf{F} : \mathbb{R}^{d_s} \rightarrow \mathbb{R}^{d_z}$ which projects a sentence representation to a latent representation, and a decoding network $\mathbf{G} : \mathbb{R}^{d_z} \rightarrow \mathbb{R}^{d_s}$ which projects the two spaces in reverse. $\boldsymbol{s}_{pp}$ denotes the feature vector derived from the preliminary prediction, which has the same dimension as ending vectors $\boldsymbol{s}^1_{end}$ or $\boldsymbol{s}^2_{end}$.

The encoder-decoder structure offers the flexibility for incorporating external information during the refined prediction. As it will be shown in Eq. 16∼17, the structure can include additional hint information in the decoding network.

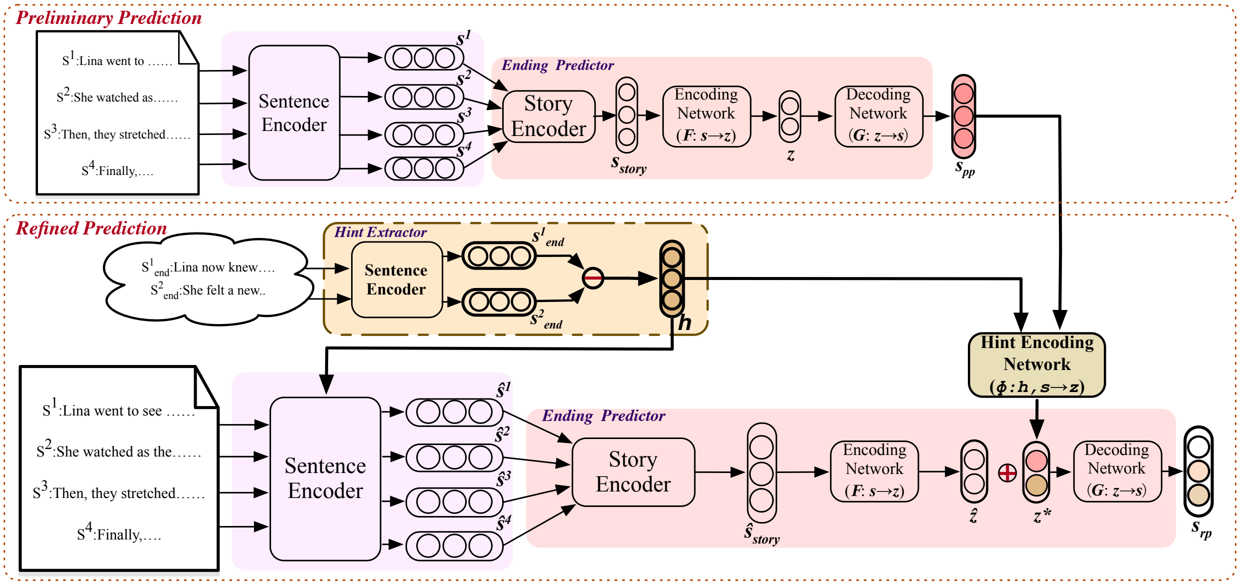

Fig. 2. HintNet applies a two-stage procedure: it anticipates a feature vector $\boldsymbol{s}_{pp}$ during the preliminary prediction and derives a refined vector $\boldsymbol{s}_{rp}$ by utilizing the hint vector $\boldsymbol{h}$ and the preliminary prediction result $\boldsymbol{s}_{pp}$. The hint vector $\boldsymbol{h}$ in the hint extractor encodes the difference between two candidate endings, and it is used to update the representation of the context sentences $\hat{\boldsymbol{s}}^i$, further to update the latent code $\hat{\boldsymbol{z}}$. The hint encoding network takes as input the predicted vector $\boldsymbol{s}_{pp}$ and the hint vector $\boldsymbol{h}$ to obtain a new code $\boldsymbol{z}^*$. The final vector $\boldsymbol{s}_{rp}$ is computed from $\hat{\boldsymbol{z}}$ and $\boldsymbol{z}^*$ by the decoding network. The modules with an identical name share the same parameters.

Furthermore, we can flexibly compare different model settings and hint representations within the structure, as shown in Section IV-C.

C. Refined Prediction

At the second stage, HintNet takes into account the hint information from two candidate endings to alleviate the evidence bias issue. In this manner, we expect to obtain a refined feature vector that may be closer to the correct ending than that in the preliminary prediction.

Hint Extractor

A useful story hint is expected to embrace the relevant clues that support a correct ending, yet exclude those which support a wrong ending, and ignore the shared part between the two endings as well. Thus, we simply define the hint vector as the subtraction of the wrong ending from the correct ending[2].

We investigate two representation schemes to obtain the hint vector: sentence-level and neural bag-of-words (NBOW) representations [42], [43].

For sentence-level representation, the hint vector is represented as follows:

$$\boldsymbol{h} = \boldsymbol{s}^c_{end} - \boldsymbol{s}^w_{end} \tag{10}$$

where $\boldsymbol{s}^c_{end}/\boldsymbol{s}^w_{end}$ denotes the sentence vector of the correct/wrong ending respectively, both produced by the sentence encoder.

For NBOW representation, the hint vector is defined as the difference at the word level, as below:

$$\boldsymbol{h}_{\mathcal{W}} = \frac{1}{|X^c_e|}\sum_{x \in X^c_e} \boldsymbol{e}(x) - \frac{1}{|X^w_e|}\sum_{x \in X^w_e} \boldsymbol{e}(x) \tag{11}$$

where $X^c_e = \{x | x \in \mathrm{S}^c_{end}\}$ is the set of words occurring in the correct ending S^c_{end}, and X^w_e is similarly defined. $\boldsymbol{e}(x)$ stands for the embedding of word x.

In addition to the above definitions, the hint representation can also be designed to incorporate other prior knowledge with the help of external resources. For example, in order to highlight words that indicate the sentiment or the event in a story [9], [22], we can represent the hint information as follows:

$$\boldsymbol{h}_{\mathcal{S}} = \frac{1}{|X^c_{\mathcal{S}}|}\sum_{x \in X^c_{\mathcal{S}}} \boldsymbol{e}(x) - \frac{1}{|X^w_{\mathcal{S}}|}\sum_{x \in X^w_{\mathcal{S}}} \boldsymbol{e}(x) \tag{12}$$

$$\boldsymbol{h}_{\mathcal{E}} = \frac{1}{|X^c_{\mathcal{E}}|}\sum_{x \in X^c_{\mathcal{E}}} \boldsymbol{e}(x) - \frac{1}{|X^w_{\mathcal{E}}|}\sum_{x \in X^w_{\mathcal{E}}} \boldsymbol{e}(x) \tag{13}$$

where $X^c_{\mathcal{S}} = \{x | x \in \mathrm{S}^c_{end} \text{ and } x \in \mathcal{S}\}$ is the set of words existing in the correct ending and a set of sentiment words $\mathcal{S}$. $\mathcal{E}$ denotes the set of event-related words. Comparison between different hint representations will be presented in Section IV-D.

It is worthwhile to mention that the labels of the two endings are available during training, however, they are unavailable during test, and we thus need some inference method, which will be introduced in Section III-E.

Refined Ending Predictor

HintNet incorporates the hint vector into both the Sentence Encoder and the Ending Predictor to obtain a better prediction.

First of all, the sentence vector is updated by attention considering the hint vector $\boldsymbol{h}$:

$$\hat{a}^i_j = (\boldsymbol{v}^i_j)^T \boldsymbol{M}\boldsymbol{h} \tag{14}$$

$$\hat{\boldsymbol{s}}^i = \sum_j^{|S^i|} \frac{exp(\hat{a}^i_j)}{\sum_k exp(\hat{a}^i_k)} \boldsymbol{v}^i_j \tag{15}$$

[2] Nevertheless, more elaborated methods for hint representations are allowed in our framework.

The updated sentence vectors $\hat{\boldsymbol{s}}^i$ are then used to update the story context's representation $\hat{\boldsymbol{s}}_{story}$ (with Eq. 7), and further to update the code $\hat{\boldsymbol{z}}$ (with Eq. 8). Thus, the representations of sentences, story, and latent code are all updated with the hint vector $\boldsymbol{h}$ during the refined prediction.

Then, a hint encoding network $\boldsymbol{\phi}$ (an MLP) integrates the hint vector $\boldsymbol{h}$ and the preliminary predicted ending vector $\boldsymbol{s}_{pp}$ to generate a latent code $\boldsymbol{z}^* \in \mathbb{R}^{d_z}$, and this code is used to obtain $\boldsymbol{s}_{rp}$ by the same decoding network $\mathbf{G}$ (in Eq. 9):

$$\boldsymbol{z}^* = \boldsymbol{\phi}(\boldsymbol{h}, \boldsymbol{s}_{pp}) \tag{16}$$

$$\boldsymbol{s}_{rp} = \mathbf{G}(\hat{\boldsymbol{z}} + \boldsymbol{z}^*) \tag{17}$$

D. Objective Function

We finally choose an appropriate ending whose sentence vector is closer to $\boldsymbol{s}_{rp}$, measured by cosine similarity:

$$k^* = \arg\max \cos(\boldsymbol{s}_{rp}, \boldsymbol{s}^k_{end}) \quad k = 1, 2 \tag{18}$$

where $\boldsymbol{s}^k_{end}$ is the vector of a candidate ending.

The model is encouraged if the predicted vector is closer to the correct ending than to the wrong ending. Hinge loss is applied here. γ_p is a hyper-parameter to adjust the margin:

$$\mathcal{L}_p = \max(0, \gamma_p + \cos(\boldsymbol{s}_{rp}, \boldsymbol{s}^w_{end}) - \cos(\boldsymbol{s}_{rp}, \boldsymbol{s}^c_{end})) \tag{19}$$

E. Inference Method

As just mentioned, obtaining the hint vector requires the access to the labels of the candidate endings, which are available during training. During test, even though we do not have such labels for computing the hint vector, we still have an effective method for story ending prediction.

For each story with two candidate endings, we can formulate two hypotheses:

Hypo-I: suppose the first candidate is correct (thus $\boldsymbol{h}^I = \boldsymbol{s}^1_{end} - \boldsymbol{s}^2_{end}$, according to Eq. 10);

Hypo-II: suppose the second candidate is correct (thus $\boldsymbol{h}^{II} = \boldsymbol{s}^2_{end} - \boldsymbol{s}^1_{end}$).

Since the hint vectors are different in the above hypotheses, we will obtain different resulting vectors $\boldsymbol{s}^I_{rp}$/$\boldsymbol{s}^{II}_{rp}$ for the two hypotheses. It is reasonable that, if Hypo-I holds, we can expect that $\cos(\boldsymbol{s}^I_{rp}, \boldsymbol{s}^1_{end}) > \cos(\boldsymbol{s}^I_{rp}, \boldsymbol{s}^2_{end})$; otherwise, $\cos(\boldsymbol{s}^I_{rp}, \boldsymbol{s}^1_{end}) < \cos(\boldsymbol{s}^I_{rp}, \boldsymbol{s}^2_{end})$. A similar rule applies to Hypo-II. The final decision is made by accepting the hypothesis which can obtain a larger value in the two scores: $\cos(\boldsymbol{s}^I_{rp}, \boldsymbol{s}^1_{end}) - \cos(\boldsymbol{s}^I_{rp}, \boldsymbol{s}^2_{end})$ and $\cos(\boldsymbol{s}^{II}_{rp}, \boldsymbol{s}^2_{end}) - \cos(\boldsymbol{s}^{II}_{rp}, \boldsymbol{s}^1_{end})$.

To assure that the inference method will accept the correct hypothesis and reject the wrong hypothesis at the same time, we design an additional hinge loss to train HintNet:

$$\begin{aligned} \mathcal{L}_w &= \max(0, \gamma_w + \cos(\bar{\boldsymbol{s}}_{rp}, \boldsymbol{s}^w_{end}) - \cos(\bar{\boldsymbol{s}}_{rp}, \boldsymbol{s}^c_{end})) \\ \mathcal{L} &= \lambda_p \mathcal{L}_p + \lambda_w \mathcal{L}_w \end{aligned} \tag{20}$$

where $\bar{\boldsymbol{s}}_{rp}$ is obtained with a wrong hint $\bar{\boldsymbol{h}} = \boldsymbol{s}^w_{end} - \boldsymbol{s}^c_{end}$. $\mathcal{L}_p$ (Eq. 19) enforces HintNet to predict a vector closer to the correct ending when the hint is computed in a correct way. Therefore, HintNet outputs the result that is accordant with the correct hypothesis. Meanwhile, when the hint vector is computed as $\bar{\boldsymbol{h}} = \boldsymbol{s}^w_{end} - \boldsymbol{s}^c_{end}$ (a wrong hypothesis assuming that S^w_{end} is correct), $\mathcal{L}_w$ drives HintNet to obtain an ending vector $\bar{\boldsymbol{s}}_{rp}$ distant from $\boldsymbol{s}^w_{end}$. By doing this, the inference method can reject the wrong hypothesis since we enforce that $\cos(\bar{\boldsymbol{s}}_{rp}, \boldsymbol{s}^w_{end}) < \cos(\bar{\boldsymbol{s}}_{rp}, \boldsymbol{s}^c_{end})$ in $\mathcal{L}_w$. γ_w is a hyper-parameter to adjust the margin. λ_p and λ_w are hyper-parameters to balance two loss functions.

IV. Experiments

A. Experimental Settings and Baselines

The corpus comes from the development set and the test set of the original Story Cloze Test dataset, similar to recent works [8], [9], [22]. Each instance consists of a four-sentence story with two candidate endings. We took 10% of the development set for validation and the rest for training. The standard test set is for evaluating models.

We used Adam [44] for optimization with a learning rate of 0.001 and a mini-batch size of 64. We performed grid search to tune hyper-parameters from $\lambda \in \{1, 2, 5\}$ and $\gamma \in \{0.1, 0.2, 0.5, 1\}$. They were finally set as: $\{\lambda_p = 2.0, \lambda_w = 1.0\}$ and $\{\gamma_p = 0.2, \gamma_w = 0.2\}$. The dimension of word vector is 300, and the dimension of hidden states in both the sentence encoder and story encoder is 128. The latent variable z has a dimension of 64. The function F, G, Φ were implemented by feed-forward networks with a single hidden layer of size 128, 128, 256, respectively. Dropout layers are applied before all linear layers with a dropout rate of 0.5. We adopted the GloVe word embeddings [45] and kept them fixed during training. For those words which have no pre-trained GloVe embeddings, we initialized them randomly and treated them as trainable variables. All parameters are regularized by the L_2 norm. We repeated each experiment five times and took the average of the results as the reported performance in following sections.

We compared our model with recent neural models and feature-based models[3]:

BinaryC [26] uses skip thought embeddings and encodes the entire context using a GRU to train a binary classifier to determine if an ending is correct or wrong.

DSSM [46] measures the cosine similarity between the vector representation of the context and that of an ending.

CGAN [25] encodes each ending and the story by GRUs and computes an entail score. This work also employs the generative adversarial networks (GANs) to generate fake candidate endings to augment the training data.

FES-LM [10] establishes a neural language model built upon frames, entities and sentiments, and then utilizes the conditional probability of the ending sentence given the context as features to determine the correct ending.

HBiLSTM [8] uses an LSTM to encode the word sequence into a sentence vector and another LSTM to encode the sentence sequence into a context vector. The vector of input candidate ending is used to compute the attention over words to obtain modified context sentence vectors. The final score

[3] For fair comparison, we did not include recent state-of-the-art results achieved by SeqMANN (84.7%), FTLM (86.5%), and Chen (87.6%) which utilized other pre-trained models.

TABLE I

ACCURACY OF HINTNET COMPARED WITH BASELINES. HINTNET IS SIGNIFICANTLY BETTER THAN BASELINES WITH P-VALUE<0.01(†) OR P-VALUE<0.05(*).

Model	Accuracy
BinaryC	0.672†
DSSM	0.585†
CGAN	0.609†
FES-LM	0.623†
HBiLSTM	0.747†
UW	0.752†
Multi Knowledge Reasoning (MKR)	0.670†
Hidden Coherence Model (HCM)	0.776*
HintNet	**0.792**

is output by an MLP whose input is the concatenation of the candidate's vector and the context vector.

UW [21] trains a logistic regression model which uses language model features and stylistic features including length, word n-grams, and character n-grams.

Multi Knowledge Reasoning (MKR) [9] selects inference rules for a certain context using attention mechanism, and measures the reasoning distance from the context to an ending by summarizing the costs of all possible inference rules.

Hidden Coherence Model (HCM) [22] trains a logistic regression model which uses features considering the event-sequence, the sentiment-trajectory, and the topic-consistency of the story.

B. Overall Performance

The experimental results in Table I show that our model outperforms all the baselines. Compared with the neural models (BinaryC, DSSM, CGAN, FES-LM, HBiLSTM), HintNet achieves a higher accuracy. Compared to the feature-based models (UW, MKR, HCM), HintNet has not only better performance but also advantages in terms of no feature engineering and less dependence on external resources. An additional significance test in terms of accuracy (p-test, [47]) shows that our system is significantly better than HCM (p-value=0.044) and other baselines (p-value<0.01). More details for the significance tests are presented in the appendix.

Table II presents some cases where HBiLSTM selects a wrong ending but HintNet predicts correctly. For each instance, we investigated the word attentions regarding the wrong ending (the top two words ranked by attention weights in each sentence are colored in red), which **visualizes the biased evidence**. As it can be seen, a wrong ending can also gather sufficient evidence with the matching-based model. Moreover, words in the biased evidence play important roles in narrating the story, which may mislead matching-based models severely. For example, in the first story, the wrong ending and the context share the same protagonists (*'my friends'* and *'I'*), the same action (*'invite'*) and the same sentiment (*'fun'*).

Table III show the attention results of HintNet for same cases. Red-colored words in the story context have largest attentions regarding the intermediate sentence vector in the preliminary prediction stage (Eq. 4 – Eq. 6) or the hint vector in the refined prediction stage (Eq. 14 – Eq. 15). When first read the context without considering any candidate endings, the self-attention avoid the evidence bias, say, not emphasize the wrong evidence *'fun'* in the above example. During refined prediction, key information to select the correct ending is emphasized by the hint vector, like *'last Saturday'*, *'terribly'* and *'broke'* to infer *'stay home next weekend'*. Moreover, distracting words like *'love'* and *'fun'* are assigned less attention in refined prediction. These intermediate results coincide with the motivation behind the hint vector (Eq. 10).

The examples show that analyzing each candidate ending separately suffers from the issue of evidence bias, particularly when the wrong ending is also highly plausible. In this circumstance, HintNet is more likely to make an accurate prediction. During anticipation, HintNet only reads the context which alleviates evidence bias. During refinement, the hint carries pairwise information which supports the correct ending and rejects the wrong one meanwhile. It is therefore easier to exclude biased evidence.

C. Impact of the Hint Information

We conducted experiments with different model structures and settings in HintNet to evaluate the influence of the hint information.

We experimented with a simple model, a one-layer feed-forward network, to test whether the hint information will help story ending prediction in even very simple structures. This simple model has three settings:

Con-Single is fed with the concatenation of the story context and a **Single** ending, $[\boldsymbol{s}_{story}; \boldsymbol{s}^1_{end}]$ or $[\boldsymbol{s}_{story}; \boldsymbol{s}^2_{end}]$, separately.

Con-Two inputs the concatenation of a story context and the two endings, $[\boldsymbol{s}_{story}; \boldsymbol{s}^1_{end}; \boldsymbol{s}^2_{end}]$.

Con-Hint inputs the concatenation of the story context and the hint information, $[\boldsymbol{s}_{story}; \boldsymbol{h}]$.

The sentence vectors and hint vectors are calculated the same as HintNet. The feed-forward network outputs a feature vector and ending selection is made by comparing cosine similarity between a candidate ending and the predicted feature vector, the same as HintNet.

The results in the first block of Table IV reveal the following observations:

- Con-Two is better than Con-Single, indicating that considering the two candidates simultaneously improves ending prediction.
- Con-Hint outperforms other two models, verifying that the hint information is useful in ending prediction, even with simple model structures.
- These baselines are worse than HintNet, which implies that it is vital to devise a proper structure to utilize the hint information, and demonstrates the effectiveness of our proposed two-stage procedure.

We experimented with different settings of HintNet to further investigate the effect of the hint information and the necessity of the proposed two-stage procedure, i.e., firstly preliminary prediction and secondly refined prediction:

HintNet (w/o Re-Prediction): Predict an ending by $\boldsymbol{s}_{pp}$, i.e., without refined prediction module.

TABLE II
Story examples on which HBiLSTM fails to select the correct ending but HintNet succeeds. Red-colored words in context have largest attention weights regarding the wrong ending in HBiLSTM, which reveals the issue of **evidence bias**.

Story	Wrong Ending	Correct Ending
My friends all love to go to the club to dance. They think it's a lot of **fun** and always **invite**. **I** finally decided **to** tag along last Saturday. I **danced** terribly and broke a **friends**'s toe.	My friends decided to keep inviting me out as I am so much fun.	The next weekend, I was asked to please stay home.
Kay loved ice **cream**. She visited Cold Stone and discovered they had cinnamon **ice cream**. She **ate** cinnamon ice **cream** every week for 3 months. One day she returned and **was** told the ice **cream was** no longer sold.	Kay was relieved because she had moved on to peppermint.	Kay was crushed!
Dave walked into **the** grocery store. He was going there **to buy** his favorite energy drink. He only had enough money to **buy one** can. He reached **the** aisle **and** what he saw made him smile.	Dave bought an entire case.	They were on sale.

TABLE III
Story examples where red-colored words in the story context have largest attention weights regarding the intermediate sentence vector (left) or the hint vector (right).

Attention-weighted context in preliminary prediction	Attention-weighted context in refined prediction
My friends all **love** to go to the club to **dance**. They **think** it's a **lot** of fun and always invite. **I** finally decided to tag along last **Saturday**. I **danced terribly** and broke a friends's toe.	My friends all love to **go** to the **club** to dance. **They think** it's a lot of fun and always invite. I finally decided to tag along **last Saturday**. I danced **terribly** and **broke** a friends's toe.
Kay loved **ice cream**. She visited **Cold** Stone and discovered they had cinnamon **ice** cream. She ate cinnamon ice cream every **week** for 3 **months**. **One** day she returned and was told the ice cream was **no** longer sold.	Kay loved **ice cream**. She visited Cold Stone and discovered they **had cinnamon** ice cream. She **ate cinnamon** ice cream every week for 3 months. One day she returned and was told the ice cream was **no longer** sold.
Dave walked into the grocery store. He was going there to **buy** his favorite energy **drink**. He **only had** enough money to buy one can. He reached the aisle and what he saw made him **smile**.	Dave walked into the **grocery store**. He **was** going there to **buy** his favorite energy drink. He **only** had enough money to buy one **can**. He **reached** the aisle and what he saw made him **smile**.

TABLE IV
Accuracy of simple MLP-models (Con-*) and HintNet with different settings.

Model	Accuracy
Con-Single	0.664
Con-Two	0.687
Con-Hint	0.711
HintNet(w/o Re-Prediction)	0.650
HintNet(w/o Hint)	0.651
HintNet(w/o Pre-Prediction)	0.770
HintNet(Multiple Refinement)	0.766
HintNet	**0.792**

TABLE V
Accuracy of HintNet with different hint representations.

Hint Repr.	Hint Source	Accuracy
$\boldsymbol{h}$	**Full sentence**	**0.792**
$\boldsymbol{h}_{\mathcal{W}}$	All words	0.728
$\boldsymbol{h}_{\mathcal{S}}$	Only sentiment words	0.692
$\boldsymbol{h}_{\mathcal{E}}$	Only event-related words	0.705
$\boldsymbol{h}_{\mathcal{S}+\mathcal{E}}$	Sentiment and event-related words	0.718

HintNet (w/o Hint): No hint is used in the structure where $\boldsymbol{z}^* = \phi(\boldsymbol{0}, \boldsymbol{s}_{pp})$, compared to Eq. 16, and $\boldsymbol{s}^i$ is not updated to $\hat{\boldsymbol{s}}^i$.
HintNet (w/o Pre-Prediction): Predict an ending without considering the preliminary prediction result by setting $\boldsymbol{z}^* = \phi(\boldsymbol{h}, \boldsymbol{0})$, compared to Eq. 16.
HintNet (Multiple Refinement): The refined prediction is executed twice, i.e., $\boldsymbol{z}_1^* = \phi(\boldsymbol{h}, \boldsymbol{s}_{pp})$ and $\boldsymbol{z}_2^* = \phi(\boldsymbol{h}, \boldsymbol{s}_{rp})$.

The results in the second block of Table IV reveal the following observations:

- The low accuracy of HintNet (w/o Re-Prediction) (0.650 vs. 0.792) and HintNet (w/o Hint) (0.651 vs. 0.792) reveal that it is difficult to perform ending prediction only with the story context.
- The comparison to HintNet (w/o Pre-Prediction) (0.770 vs. 0.792) shows that an preliminary prediction is beneficial as it generates an anticipated vector for a possible ending. It is useful when combined with the hint information.
- The accuracy of multiple refinement drops compared with single refinement because the model is trained to overly depend on the hint information. Thus, abuse of the hint information damages the performance of story ending prediction.

D. Impact of Different Hint Representations

We evaluated the impact of different representations of the hint on the performance, as described in Section III-C, namely sentence-level representation ($\boldsymbol{h}$), and neural bag-of-words (NBOW) representations ($\boldsymbol{h}_{\mathcal{W}}$, $\boldsymbol{h}_{\mathcal{S}}$, and $\boldsymbol{h}_{\mathcal{E}}$). Moreover, we investigated another NBOW representation $\boldsymbol{h}_{\mathcal{S}+\mathcal{E}}$, considering both the sentiment and event-related words in the hint representation.

TABLE VI
PERFORMANCE CHANGES WHEN DIFFERENT TYPES OF WORDS ARE REMOVED FROM THE ENDINGS WHEN COMPUTING HINT VECTORS.

Removed Tag	Accuracy	Correct Ending	Wrong Ending
VB*	↓ 8.4%	Sam ~~liked(VBD)~~ it. Franny learned to ~~examine(VB)~~ her prejudices.	Sam ~~hated(VBD)~~ it. Franny ~~ended(VBD)~~ up ~~getting(VBG) deported(VBN)~~.
JJ	↓ 6.2%	Kelly was so ~~happy(JJ)~~ to finally beat it. Josh got ~~sick(JJ)~~.	Kelly was ~~mad(JJ)~~ about that. Josh thought the pie was ~~deicious(JJ)~~.
NN/NNS	↓ 7.7%	She loved her new ~~phone(NN)~~. My ~~friends(NNS)~~ stopped playing to help me off the ~~field(NN)~~	Amy spent all of her ~~money(NN)~~ on ~~clothes(NNS)~~. I got back up to finish the ~~game(NN)~~
NNP/PRP	↓ 3.1%	~~He(PRP)~~ decided to run away from home. ~~His(PRP)~~ dad's teasing makes ~~Henry(NNP)~~ feel bad.	~~Tommy(NNP)~~ then bought a new car. ~~Henry(NNP)~~ wished ~~he(PRP)~~ looked like the handsome mailman.

To obtain sentiment-specific hint representation, we used the sentiment vocabulary[4] from [48] which includes 2,006 positive and 4,783 negative words. To obtain event-specific hint representation, we used the NLTK toolkit for part-of-speech tagging and collected all verbs and nouns in the Story Cloze dataset to construct an event vocabulary.

The results in Table V demonstrate that:

- NBOW hint representations are worse than the sentence-level representation since the former are too shallow to capture the deep meaning of a sentence in story comprehension. Statistics in Section IV-F report that most stories need deep semantic understanding besides words matching, also encouraging the sentence-level representation.
- Jointly considering sentiment or event-related words in NBOW representation can improve story comprehension compared with separate use of such resources,which is in line with [9], [22].

E. Connections between Hint and Words

We studied the connections between the hint vector and the surface words in the endings to reveal what is captured by the hint vector. We compared the performance when a particular type of words are removed from the endings. The sentence-level hint representation is applied to the modified ending, and the final decision is made by comparing s_{rp} to the vectors of untouched candidate endings.

The word type is decided by the part-of-speech tag of a word, using the NLTK[5] toolkit. We compared four word types: verbs (VB*), adjectives (JJ), nouns (NN/NNS), and proper nouns plus personal pronouns (NNP/PRP).

Table VI presents the results with some exemplar endings. We had the following observations:

- **Verbs (VB*)** play the most important roles in hint representation where the performance drops mostly without verbs, possibly because verbs usually express the key information about the action, event, or major logic of a reasonable story.
- **Adjectives (JJ)** are important for obtaining the hint information since such words usually express strong sentiment and emotion, which is crucial to decide a coherent ending. It is in line with [9], [22].
- **Noun phrases (NN/NNS)** are important for hint representation as such words are usually the object of an action, representing the candidates' differences in the recipients of an action, or the effect of an event.
- **Proper nouns and personal pronouns (NNP/PRP)** tend to be less important for hint representation possibly due to the fact that these words are usually the subject of a sentence, representing the shared information between the two candidates. For instance, the protagonist typically appears in both candidate endings.

We sampled 150 stories from the test set randomly and annotated them manually to investigate the type distribution of key words. For each instance, annotators are asked to select several words that distinguished a correct ending from a wrong candidate. Fig. 3 shows the distribution of part-of-speech tags for these keywords[6].

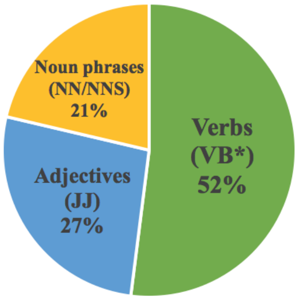

Fig. 3. The distribution of part-of-speech tags of manually annotated keywords in the 150 sampled stories. No proper noun or personal pronoun is selected as keyword by annotators.

Results show that the distribution statistics are consistent with the results in Table VI. Verbs are the most crucial keywords for selecting a correct ending. Adjectives and noun phrases are also important. Note that no proper nouns nor personal pronouns are selected as keywords by human annotators, indicating that such words are not important for ending selection (smallest performance drop when removing such words, see Table VI), since both candidates follow up the story context by sharing characters [2].

[4] http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar

[5] http://www.nltk.org

[6] We do not count the keywords whose POS tags are not one of the four types we compared. The proportion of excluded keywords is less than 5%.

TABLE VII
REQUIREMENTS OF SUCCESSFUL ENDING PREDICTION AND CORRESPONDING STORY EXAMPLES.

Requirement of Successful Ending Prediction	**Context**	**Candidate Endings (Correct in bold)**
Surface Word Matching **(31%)**	Aaron's girlfriend asked him to come over for dinner. She said she was making his favorite, chicken alfredo. Aaron was very excited she wanted to cook, but he hated alfredo. She must have mixed up his words when he told her his least favorite.	**Aaron suggested he help her cook another meal instead.** (*'cook' is enough*)
		Aaron broke up with her.
Deep Semantic Coherence **(52%)**	Tom rolled his wagon. The wheels then fell off! Tom started crying! Tom's dad fixed the wheels.	**Tom's dad was always there for Tom.** (*fixed→helpful= be there for*)
		Tom's dad was not helpful ever.
Implicit Human Knowledge **(17%)**	Ben had a doctor's appointment. He was very scared. He never went to the doctor. He slowly stepped into the office.	**The doctor greeted him calmly and he felt better.**
		Ben was having so much fun there. (*seeing a doctor is not fun!*)

F. Case Analysis

It is observed that the current capacity of story comprehension is still far from the human-level performance. Fig. 4 shows an example for which HintNet failed to select the correct candidate. In the story context, there are many positive words such as *'date'* and *'like'*, but the correct ending describes an upset event. In order to select the correct ending in this example, the machine should not only track changes of events and their participants, but also possess the knowledge that a man can have only one lover. Only through the surface words, the model cannot do well on this example.

My dad was **dating a girl**. ***The girl*** had a friend. ***The friend*** **liked** my dad. My dad eventually **started to date the friend**.
Wrong Ending: My dad went to work. Correct Ending: ***The first girl*** was very **upset**.

Fig. 4. A story example where HintNet fails to select the correct ending.

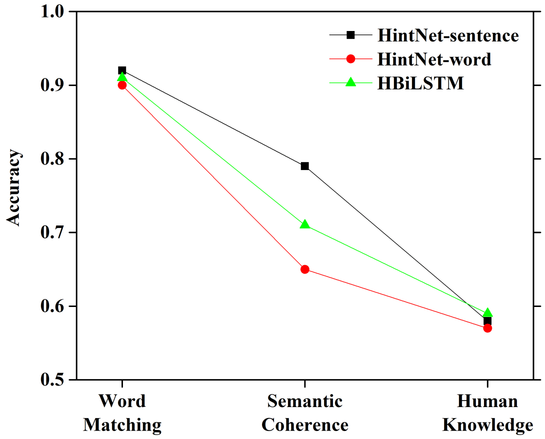

Fig. 5. Accuracy of three neural models on stories with different reasoning requirements.

We note that the difficulty level for story ending selection varies case by case. Some cases can be done by simple surface word matching, while some cases require implicit human knowledge, which is much more difficult. To this end, we sampled 150 examples from the test set randomly . Here, we broadly divide the requirements of successful ending prediction into the following categories with examples shown in Table VII:

1) **Surface Word Matching (31%)** : Words in the correct ending are more coherent with the context, which makes it easy to identify the correct ending at the lexical level.
2) **Deep Semantic Coherence (52%)**: It requires a comprehensive understanding of sentences and a deep understanding of the relationship between multiple sentences to select the correct ending.
3) **Implicit Human Knowledge (17%)**: The correct endings in these examples are hard to identify without some real-world knowledge. Even for some instances, both endings are logically reasonable but the correct one is more better in terms of aesthetics.

Fig. 5 displays the accuracy of HintNet with $\boldsymbol{h}$ (sentence-level hint representation, *HintNet-sentence*), HintNet with $\boldsymbol{h}_{\mathcal{W}}$ (word-level hint representation, *HintNet-word*) and *HBiLSTM* model on these samples:

- All the models perform closely to each other and obtain high accuracy (0.9) on the samples which require only surface word matching. That means, the easiest cases can be handled well by all the models.
- The performance difference is remarkable on the samples that need to handle semantic coherence. The accuracy of the three models varies from 0.6 to 0.8. *HintNet-sentence* is much better than *HintNet-word* since the former can capture sentence meanings better. The accuracy of *HBiLSTM* is between *HintNet-sentence* and *HintNet-word*. Both *HBiLSTM* and *HintNet-sentence* encode full sentences using a bidirectional LSTM, but *HBiLSTM* is negatively affected by the biased attention from an individual ending.
- None of the three models can predict correct endings well when implicit human knowledge is required. All the ac-

curacy scores are lower than 0.6, close to a random guess. This demonstrates the challenges for the hardest case of story comprehension, and also the future direction.

V. Conclusion And Future Work

This paper studies the evidence bias issue in story ending selection, which commonly exists in the methods where each candidate ending is matched to the story context separately. We present a two-stage neural model, HintNet, to address the evidence bias issue and improve story comprehension. HintNet firstly anticipates a feature vector which implies a possible ending, and then refines its preliminary prediction using the hint information which encodes the difference between two candidate endings. Results demonstrate HintNet outperforms best-performing baselines, thereby justifying the benefit of using hint information via simultaneous comparison.

As future work, there is still much room on using implicit or explicit knowledge for story comprehension.

Appendix A
Significance Test

We present the details of how significance tests are conducted. We apply p-test [47] to compare the performance measures which are proportions.

Here we take the best baseline system (HCM) as the reference system and other comparisons follow a similar process. The observed accuracy of our system is $p_o^a = 0.792$, and that of the reference system (HCM) is $p_o^b = 0.776$. The observed proportion of the total trials ($n^a = n^b = 1871$) is:

$$p = \frac{n^a \times p_o^a + n^b \times p_o^b}{n^a + n^b} = \frac{p_o^a + p_o^b}{2} = 0.784$$

The null hypothesis is $H_0 : p^a = p^b = p$. The alternative hypothesis is $H_1 : \ p^a > p^b$, which means our model is better than HCM.

Since the number of trials $n^a = n^b = 1871$, the observed value for one-sided test statistic is computed as:

$$\begin{aligned} z_o &= \frac{p_o^a - p_o^b}{\sqrt{2p(1-p)/(n^a + n^b)}} \\ &= \frac{0.792 - 0.776}{\sqrt{2 * 0.784 * 0.216/3742}} \\ &= \frac{0.016}{0.0095} \\ &= 1.68 \end{aligned}$$

Thus, the $p - value$ can be computed using the standard normal distribution:

$$\begin{aligned} P(Z \geq z_o) &= 1 - \Phi(1.68) \\ &= 1 - 0.9554 \\ &= 0.0446 \end{aligned} \tag{21}$$

Since the $p - value$ is smaller than $\alpha = 0.05$, we can reject the null hypothesis and accept that our model is significantly better than HCM at the significance level of $\alpha = 0.05$.

Acknowledgment

This work was supported by the National Key R&D Program of China (Grant No. 2018YFC0830200), and partly by the National Science Foundation of China (Grant No.61876096/61332007). The authors would like to thank all the workers who help us to annotate sampled data. Mantong Zhou would like to acknowledge Dr.Yao Xiao for fruitful discussions.

References

[1] I. Mani, "Computational modeling of narrative," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 3, p. 142, 2012.

[2] N. Mostafazadeh, N. Chambers, X. He, D. Parikh, D. Batra, L. Vanderwende, P. Kohli, and J. Allen, "A corpus and cloze evaluation for deeper understanding of commonsense stories," in *NAACL*, 2016, pp. 839–849.

[3] M. Bugert, Y. Puzikov, A. Rckl, J. Eckle-Kohler, T. Martin, E. Martinez Camara, D. Sorokin, M. Peyrard, and I. Gurevych, "LSDSem 2017: Exploring Data Generation Methods for the Story Cloze Test," in *Proceedings of the 2nd Workshop on LSDSem*. Valencia, Spain: Association for Computational Linguistics, April 2017, pp. 56–61.

[4] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, November 2016, pp. 2383–2392.

[5] K. M. Hermann, T. Kocisk, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," *neural information processing systems*, pp. 1693–1701, 2015.

[6] M. Richardson, C. J. Burges, and E. Renshaw, "Mctest: A challenge dataset for the open-domain machine comprehension of text," in *Empirical Methods in Natural Language Processing*, 2013, pp. 193–203.

[7] F. Hill, A. Bordes, S. Chopra, and J. Weston, "The goldilocks principle: Reading children's books with explicit memory representations," *international conference on learning representations*, 2016.

[8] Z. Cai, L. Tu, and K. Gimpel, "Pay attention to the ending:strong neural baselines for the roc story cloze task," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 616–622. [Online]. Available: http://aclweb.org/anthology/P17-2097

[9] H. Lin, L. Sun, and X. Han, "Reasoning with heterogeneous knowledge for commonsense machine comprehension," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 2032–2043. [Online]. Available: https://www.aclweb.org/anthology/D17-1216

[10] H. Peng, S. Chaturvedi, and D. Roth, "A joint model for semantic sequences: Frames, entities, sentiments," in *Conference on Computational Natural Language Learning (CoNLL)*. Association for Computational Linguistics, 2017, pp. 173–183.

[11] R. C. Schank and R. P. Abelson, "Scripts, plans, goals and understanding: An inquiry into human knowledge structures." *American Journal of Psychology*, vol. 92, no. 1, p. 176, 1977.

[12] N. Chambers and D. Jurafsky, "Unsupervised learning of narrative event chains," in *ACL 2008, Proceedings of the Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, Usa*, 2008, pp. 789–797.

[13] N. Chambers, "Event schema induction with a probabilistic entity-driven model," pp. 1797–1807, 01 2013.

[14] M. Regneri, A. Koller, and M. Pinkal, "Learning script knowledge with web experiments," in *Meeting of the Association for Computational Linguistics*, 2010, pp. 979–988.

[15] N. Balasubramanian, S. Soderland, Mausam, and O. Etzioni, "Generating coherent event schemas at scale," in *Empirical Methods in Natural Language Processing*, 2013.

[16] K. Pichotta and R. Mooney, "Statistical script learning with multi-argument events," pp. 220–229, 01 2014.

[17] R. Rudinger, P. Rastogi, F. Ferraro, and B. V. Durme, "Script induction as language modeling," in *Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1681–1686.

[18] R. Soricut and D. Marcu, "Discourse generation using utility-trained coherence models," in *Coling/acl on Main Conference Poster Sessions*, 2006, pp. 803–810.

[19] A. Louis and A. Nenkova, "A coherence model based on syntactic patterns," in *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 1157–1168.

[20] K. Pichotta and R. J. Mooney, "Statistical script learning with multi-argument events," in *European Chapter of the Association for Computational Linguistics*, Gothenburg, Sweden, April 2014, pp. 220–229.

[21] R. Schwartz, M. Sap, I. Konstas, L. Zilles, Y. Choi, and N. A. Smith, "Story cloze task: Uw nlp system," in *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, April 2017, pp. 52–55.

[22] S. Chaturvedi, H. Peng, and D. Roth, "Story comprehension for predicting what happens next," in *Proceedings of the Conference on EMNLP*, September 2017, pp. 1603–1614.

[23] H. Peng and D. Roth, "Two discourse driven language models for semantics," *ACL*, 2016.

[24] N. Mostafazadeh, L. Vanderwende, W.-t. Yih, P. Kohli, and J. Allen, "Story cloze evaluator: Vector space representation evaluation by predicting what happens next," in *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 24–29. [Online]. Available: http://anthology.aclweb.org/W16-2505

[25] B. Wang, K. Liu, and J. Zhao, "Conditional generative adversarial networks for commonsense machine comprehension," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, (IJCAI)*, 2017, pp. 4123–4129.

[26] M. Roemmele, S. Kobayashi, N. Inoue, and A. Gordon, "An rnn-based binary classifier for the story cloze test," in *The Workshop on Linking MODELS of Lexical*, 2017, pp. 74–80.

[27] S. Srinivasan, R. Arora, and M. Riedl, "A simple and effective approach to the story cloze test," *CoRR*, vol. abs/1803.05547, 2018.

[28] Q. Li, Z. Li, J.-M. Wei, Y. Gu, A. Jatowt, and Z. Yang, "A multi-attention based neural network with external knowledge for story ending predicting task," in *International Conference on Computational Linguistics*, 2018, pp. 1754–1762.

[29] "Improving language understanding by generative pre-training," 2018. [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pd

[30] J. C. Chen, Jiaao and Z. Yu., "Incorporating structured commonsense knowledge in story completion." in *AAAI*, 2019.

[31] A. Naik, A. Ravichander, N. Sadeh, C. Rose, and G. Neubig, "Stress test evaluation for natural language inference," in *27th International Conference on Computational Linguistics*. Association for Computational Linguistics, 2018, pp. 2340–2353.

[32] G. Mendez, P. Gervas, and C. Leon, "A model of character affinity for agent-based story generation," in *International Conference on Knowledge, Information and Creativity Support Systems*, 2014.

[33] M. O. Riedl and C. Len, "Toward vignette-based story generation for drama management systems," in *Intelligent Technologies for Interactive Entertainment*, 2008.

[34] Z. Li, X. Ding, and T. Liu, "Generating reasonable and diversified story ending using sequence to sequence model with adversarial training," in *International Conference on Computational Linguistics*. Association for Computational Linguistics, 2018, pp. 1033–1043.

[35] J. Guan, Y. Wang, and M. Huang, "Story ending generation with incremental encoding and commonsense knowledge," 2019.

[36] A. Fan, M. Lewis, and Y. Dauphin, "Hierarchical neural story generation," in *ACL*, 2018, pp. 889–898.

[37] B. Li, S. Lee-Urban, G. Johnston, and M. O. Riedl, "Story generation with crowdsourced plot graphs," in *AAAI*, 2013, pp. 598–604.

[38] V. Soo, C. Lee, and T. Chen, "Generate believable causal plots with user preferences using constrained monte carlo tree search," in *AAAI*, 2016, pp. 218–224.

[39] Y. Ji, C. Tan, S. Martschat, Y. Choi, and N. A. Smith, "Dynamic entity representations in neural language models," in *EMNLP*, 2017, pp. 1830–1839.

[40] E. Clark, Y. Ji, and N. A. Smith, "Neural text generation in stories using entity representations as context," in *NAACL*, 2018, pp. 1631–1640.

[41] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[42] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daum III, "Deep unordered composition rivals syntactic methods for text classification," *Conference: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 1681–1691, 01 2015.

[43] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*.

[44] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *international conference on learning representations*, 2015.

[45] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.

[46] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proceedings of international conference on Conference on information & knowledge management*. ACM, 2013, pp. 2333–2338.

[47] Y. Yang, "A re-examination of text categorization methods," in *International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, pp. 42–49.

[48] M. Hu and B. Liu, "Mining and summarizing customer reviews," *ACM SIGKDD-2004*, pp. 168–177, 2004.

Mantong Zhou received the B.E. degree from Tsinghua University, Beijing, China, in 2015. she is a PhD student at the Department of Computer Science and Technology, Tsinghua University. Her research interests include question answering and machine comprehension.

Minlie Huang received the Ph.D. degree from Tsinghua University, in 2006. He is currently an associate Professor at the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His research interests include deep/reinforcement learning and natural language processing. He has published 60+ papers in premier conferences and journals (ACL, IJCAI, AAAI, KDD, SIGIR, WWW, etc.). He won IJCAI 2018 distinguished paper award, CCL 2018 best demo award, NLPCC 2015 best paper award, and Hanvon Youngth Innovation Award in 2018. His work on emotional chatting machines was reported by MIT Technology Review, the Guardian, nVIDIA, and other mass media.

Xiaoyan Zhu received the bachelor degree at University of Science and Technology Beijing in 1982, master degree at Kobe University in 1987. and Ph. D. degree at Nagoya Institute of Technology, Japan in 1990. She is currently a Professor at the Department of Computer Science and Technology, Tsinghua University, Beijing, China. Her research interests include intelligent information processing, machine learning, natural language processing, quanry and answering system and Bioinformatics. She has authored more than 100 peer-reviewed articles in leading international conferences (SIGKDD, IJCAI, AAAI, ACL) and journals (TOIS, Bioinformatics, Genome Biology)