Unsupervised Story Comprehension with Hierarchical Encoder-Decoder

Bingning Wang[2], Ting Yao[2], Qi Zhang[2], Jingfang Xu[2]
Zhixing Tian[1], Kang Liu[1], Jun Zhao[1]
1. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
Beijing, 100190, China
2. Sogou Inc.
Beijing, 100084, China
wangbingning,yaoting,qizhang,xujingfang@sogou-inc.com
zhixing.tian,kliu,jzhao@nlpr.ia.ac.cn

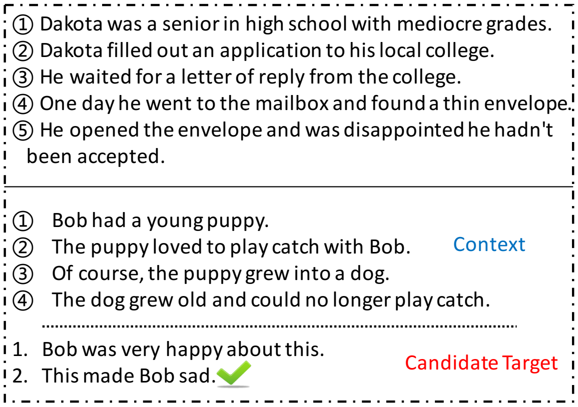
ABSTRACT

Commonsense understanding is a long-term goal of natural language processing yet to be resolved. One standard testbed for commonsense understanding is *Story Cloze Test* (SCT) [22], In SCT, given a 4-sentences story, we are expected to select the proper ending out of two proposed candidates. The training set in SCT only contains unlabeled stories, previous works usually adopt the small labeled development data for training, which ignored the sufficient training data and, essentially, not reveal the commonsense reasoning procedure. In this paper, we propose an unsupervised sequence-to-sequence method for story reading comprehension, we only adopt the unlabeled story and directly model the context-target inference probability. We propose a loss-reweight training strategy for the seq-to-seq model to dynamically tuning the training process. Experimental results demonstrate the advantage of the proposed model and it achieves the comparable results with supervised methods on SCT.

KEYWORDS

Machine Comprehension, Unsupervised Learning

ACM Reference Format:
Bingning Wang[2], Ting Yao[2], Qi Zhang[2], Jingfang Xu[2] and Zhixing Tian[1], Kang Liu[1], Jun Zhao[1]. 2019. Unsupervised Story Comprehension with Hierarchical Encoder-Decoder. In *The 2019 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '19), October 2–5, 2019, Santa Clara, CA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3341981.3344227



① Dakota was a senior in high school with mediocre grades.
② Dakota filled out an application to his local college.
③ He waited for a letter of reply from the college.
④ One day he went to the mailbox and found a thin envelope.
⑤ He opened the envelope and was disappointed he hadn't been accepted.

① Bob had a young puppy.
② The puppy loved to play catch with Bob.
③ Of course, the puppy grew into a dog.
④ The dog grew old and could no longer play catch.

1. Bob was very happy about this.
2. This made Bob sad.✓

Figure 1: An example of SCT, the upper half is a training instance and the lower half is a test instance.

1 INTRODUCTION

Machine comprehension (MC) of text is one of the ultimate goals in natural language processing (NLP) and artificial intelligence. However, teaching a machine to *comprehend* text is extremely challenging since comprehension involves many aspects of knowledge, such as information retrieval, fact reasoning, commonsense inference, etc. [36]. In recent years, many datasets have been proposed to evaluate the comprehension ability of a system, such as MCTest [29], SQuAD [28], MARCO [23], or cloze style datasets such as CNN/Daily Mail [13], Clicr [37] etc. However, most of these datasets are focused on factoid questions, and the reasoning abilities required to answer these questions are limited to shallow linguistic features, which makes it easy for even simple keyword matching algorithms to achieve high accuracy [35, 36]. The deeper inference ability of a system has not been thoroughly evaluated. Story Cloze Test (SCT) [22], on the contrary, is a story cloze dataset that requires deeper understanding of the document. This dataset contains many stories, each story is made up of 5 highly recapitulative sentences. The story in SCT captures a rich set of causal and temporal relations between daily events. During the test period, given a four sentences story plot (context), we must predict from two candidates that which is more likely to be inferred from the context. SCT requires reasoning with implicit commonsense knowledge, rather than matching explicit information in the text. An example of SCT is shown in Figure 1.

A main characteristic of SCT is that the training data is unlabeled which only contains the positive examples (i.e., the 5th sentence), so the traditional discriminative models are hard to apply. Nonetheless, the development set is similar to the test set that contains the human-crafted negative sentences, so previous works on this task usually adopt the small labeled development set for training [2, 6, 17, 20, 27, 31, 33]. However, only utilizing the development set may not reveal the real difficulty of the commonsense reasoning in SCT. For example, Schwartz et al. [31] find that when trained on the

ICTIR '19, October 2–5, 2019, Santa Clara, CA, USA
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6881-0/19/10...$15.00
https://doi.org/10.1145/3341981.3344227

development set, only using the target sentence (without the context) for classification could yield a very good result, which means the development set (and test set) is biased towards some linguistic features in the target sentence, the inference procedure from the context to the target has not been throughout exploited.

In this paper, rather than employing the superficial linguistic features adopted by previous methods, we focus on the commonsense inference procedure between the context and the target. So instead of using the biased development set we directly use the unlabeled stories for training. Concretely, we use a sequence-to-sequence model to *transform* the context to the target, we model the context by an encoder and then generate the target sentence word by word via a decoder. The loss is the cross-entropy between the generated word and ground truth target word. The encoder is a hierarchical model consists of two LSTMs to represent the meaning of the context by *word⇒sentence⇒document* hierarchy. The decoder is another LSTM model which is trained to maximize the likelihood of the target. During inference period, we cast the sentence classification problem as a conditional probability estimation problem, the prediction is the sentence that has higher decoding likelihood.

Nonetheless, for the seq-to-seq model, the training instances may result in the *optimization-inequality*: when the context contains more sentences, it would be more confident about what will happen in the next, and the prediction loss of the decoder should be small; when given only a little (or even no) context information, it would be less certain about the next sentence, and the corresponding decoder loss is expected to be high. To take the *hardship* of each training instance into account, we propose a novel loss-reweighted method to train the decoder. For each sentence, we use its encoder hidden representation to determine its weight in the final loss calculation. In this manner, the loss weight is tailored to the data itself.

In addition, as the proposed methods are pure unsupervised which only requires the context-target pairs. The abundant instance in other text may benefit a lot for our reasoning system. Inspired by the recent success of utilizing the large unlabeled data for natural language processing, such as BERT [9], ELMO [25] Skip-thought [16]. We extend the proposed methods on external large unlabeled data, such as BookCoprpus and Wikipedia, which serves as the pre-training step. We found that this pre-training is very useful for our application and achieves a significant improvement in the final result.

We conduct several experiments on SCT. Our hierarchical encoder-decoder obtains nearly 10 percent absolute improvements over other unsupervised methods. And when enhanced with abundant external data, our model even achieves comparable results with most supervised counterparts. The results and quality analysis reveal that:

- The proposed unsupervised models are more suitable to the context-inference problem compared with the discriminative model when it is hard to get the negative sample.
- When pre-trained on the large unlabeled dataset, we could obtain significant improvement, which means the abundant unlabeled data contains a lot of information that benefit our commonsense inference application.
- The quality of the generated sentence relies on the amount of information in the context, so devising a strategy (loss-reweight in this paper) to take each sentence weight into account is important to train the decoder.

2 METHODOLOGY

Each story in SCT contains five consecutive sentences $D = (s_1, ..., s_5)$ where each sentence s_i consists of a sequence words: $s_i = (w_{i1}, ..., w_{in})$. During inference period, given four context sentences $C = (s_1, ..., s_4)$ we should predict which candidate sentence, i.e. t_1 or t_2, is more likely to be inferred by C.

2.1 Hierarchical Encoder Decoder

2.1.1 Encoder. The encoder is a hierarchical model, which consists of a sentence level LSTM encoder to processed the words; and a document level LSTM to process the sentences.

Sentence level encoder takes the word embedding $\mathbf{w}$ as input, then process the sentences in the context forward and backward with two separate LSTMs:

$$\begin{aligned}\overrightarrow{\mathbf{h}_t} &= \overrightarrow{LSTM}(\overrightarrow{\mathbf{h}}_{t-1}, \mathbf{w}_t)\\ \overleftarrow{\mathbf{h}_t} &= \overleftarrow{LSTM}(\overleftarrow{\mathbf{h}}_{t+1}, \mathbf{w}_t)\end{aligned} \tag{1}$$

we concatenate the forward and backward representation for each word: $\mathbf{h}_t = [\overrightarrow{\mathbf{h}_t}; \overleftarrow{\mathbf{h}_t}]$. Finally, we average each word hidden representation as the sentence representation: $\mathbf{s}_i = \frac{1}{n_i}\sum_{t=1}^{n_i}\mathbf{h}_{it}$ where n_i is the number of word in ith sentence.

Document level encoder is built upon the sentence level representation to derive a global representation of the document. It processes the sentences with a uni-directional LSTM that takes the sentence representation $\mathbf{s}_i$ as input:

$$\mathbf{o}_i = LSTM(\mathbf{o}_{i-1}, \mathbf{s}_i), \quad \mathbf{c}_i = \frac{1}{i}\sum_{j=1}^{i}\mathbf{o}_j \tag{2}$$

where $i \in [1, 4]$. $\mathbf{o}_i$ is the document representation and $\mathbf{c}_i$ is the document embedding for sentence i that fed into the decoder.

2.1.2 Target Sentence Attentive Decoder. After encoding the context sentences $[s_1, ..., s_i]$ with the above two-level LSTM architecture, we use a decoder to decode the target sentence word by word. The objective of the decoder is to maximize the log likelihood of the target sentence:

$$\log P(s_{i+1}|s_{1:i}) = \sum_{t=1}^{n_{i+1}}\log P(w_t|w_{1:t-1}, s_{1:i}) \tag{3}$$

and each word probability could be calculated by:

$$\begin{aligned}\mathbf{h}_t^d &= LSTM_{decoder}(\mathbf{h}_{t-1}; w_{t-1})\\ \widehat{\mathbf{h}}_t &= [\mathbf{h}_t^d; \mathbf{c}_i], \quad \overline{\mathbf{h}}_t = tanh(\mathbf{W}_p \cdot \widehat{\mathbf{h}}_t)\\ P(w_t|w_{1:t-1}, s_{1:i}) &= \frac{\exp(\overline{\mathbf{h}}_i^T \cdot \mathbf{w}_t)}{\sum_{j=1}^{|V|}\exp(\overline{\mathbf{h}}_i^T \cdot \mathbf{w}_j)}\end{aligned} \tag{4}$$

The superscript d stands for 'decoder', $\mathbf{W}_p$ is a projection matrix to transform the decoder hidden representation into the word embedding space. $|V|$ is the vocabulary size.

However, decoding the target sentence s_{i+1} merely by the context vector $\mathbf{c}_i$ is not an elegant way, because different part of the target sentence may derived from different context sentences. In this paper, we embed the well-developed attention mechanism [19] into the

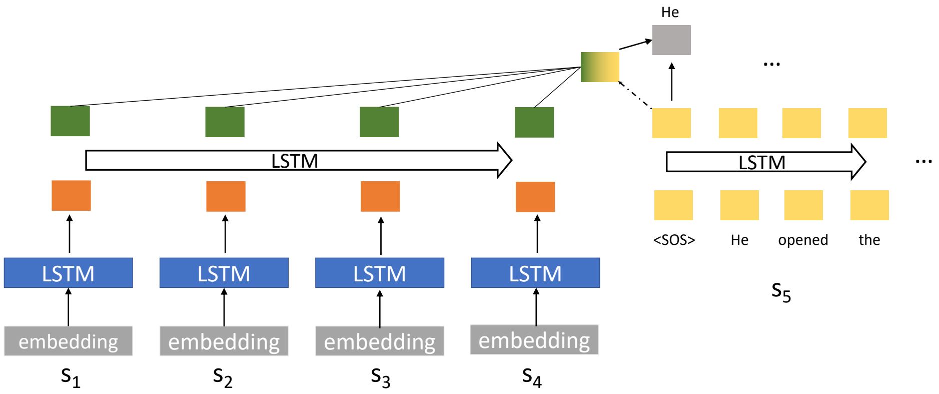


Figure 2: The whole architecture of our hierarchical encoder-decoder to model the context and target.

decoder: Instead of a fixed context representation $\mathbf{c}_i$, we use an attention-weighted context representation:

$$\begin{gathered}\alpha_j \propto \mathbf{o}_j^T \mathbf{h}_t^d, \quad \mathbf{a}_t = \sum_j^t \alpha_j \mathbf{o}_j \\ \widehat{\mathbf{h}}_t = [\mathbf{h}_t^d; \mathbf{a}_t]\end{gathered} \tag{5}$$

α_j is the attention score for sentence j respect to the current word t.

2.1.3 Loss Reweighted Training Strategy. During the decoding period, traditional encoder-decoder architecture takes the summation of each word loss as the sentence loss, which means each word takes the equal weight. However, when decoding the several opening words, we have less information, so the predictions are somewhat random. On the contrary, after predicting some words, we are confident about what to decode next. In this work, we use the word hidden representation $\mathbf{h}_t^d$ to determine its loss weight. Concretely, for a specific sentence s_i with length n_i, the loss is:

$$\begin{aligned}\beta'_{ij} &= \mathbf{w}_s^T \cdot \mathbf{h}_{ij}^d, \quad j = 1, \ldots, n_i \\ \beta_{ij} &= \frac{\exp \beta'_{ij}}{\sum_{k=1}^{n_i} \exp \beta'_{ik}} \\ \mathcal{L}_i &= -\sum_{j=1}^{n_i} n_i \cdot \beta_{ij} \cdot \log P(w_j | w_{1:j-1}, s_{1:i-1})\end{aligned} \tag{6}$$

$\mathbf{w}_s$ is the weight vector to calculate β_{ij}-the (normalized) weight for jth word in final loss summation. When set this weight to constant $\frac{1}{n_i}$, the model is reduced to traditional sum-loss scheme in Equation 3. In this manner, the loss of each word is not equal or fixed but tuned by the model i.

Document Level Loss Reweight: In SCT, each story instance contains 5 ⟨*context-target*⟩ pairs: $\{\langle s_{0:i-1}, s_i \rangle | i \in [0, 4]\}$. We use each sentence representation $\mathbf{o}_i$ to determine its loss weight in the context, thus the loss for each story is:

$$\begin{aligned}\gamma'_i &= \mathbf{w}_d^T \cdot \mathbf{o}_i \\ \gamma_i &= \frac{\exp \gamma'_i}{\sum_{k=1}^{5} \exp \gamma'_k} \\ \mathcal{L} &= \sum_{i=1}^{5} \gamma_i \cdot \mathcal{L}_i\end{aligned} \tag{7}$$

$\mathbf{w}_d$ is the document level loss weight vector. $\mathcal{L}$ is the final objective we try to minimize.

2.1.4 Inference. As the whole architecture is optimized to maximize the likelihood of the target sentence conditioned on the context, so the inference could be made by how likely the target is given the context. We select the target that has a less conditional perplexity:

$$P(t_i | s_{1:4}) = \exp \frac{1}{n_i} \sum_{j=1}^{n_i} \log P(w_j) \tag{8}$$

where $P(w_j)$ is defined in Equation 4. This criterion is similar with Schwartz et al. [31] who also proposed a language model for this task, nevertheless, our architecture is hierarchical so we directly adopt the target sentence perplexity.

2.2 Pre-training

As the whole architecture only takes the unlabeled context-target pairs as input that it could trivially enhance our model by the large amount unlabeled data, such as Wikipedia articles or fiction stories. In this paper, we choose two types of unlabeled text as pre-training resources. The first one is the BookCorpus stories [41] that contains 2662 unpublished novels from 16 categories such as Mystery Adventure or Science fiction. The second one is the Wikipedia article, which consists of the detailed description for the item in the world. This two types of external unlabeled data are complementaries to each other consists of different aspects of commonsense.

Compared to previous pre-training methods, our methods have two characteristics. 1) As the inference is made by the context-target probability in Equation 8, the pre-training and fine-tuning step are equivalent so there is no need for devising new architecture for the fine-tuned model, and even obviates the fine-tuned step that we could directly use the pre-trained model for inference. 2) Previous pre-training method for NLP mainly focuses on local information, for example, word embedding is mainly focus on word level information, and Skip-Thought, Elmo, and Bert is mainly focus on sentence-level pre-training. However, we focus on document level information which captures the long-range dependencies between sentences.

To better utilize the abundant training data, in this paper, for each sentence s_i in the unlabeled dataset, we treat its preceding n sentences, i.e. $\{s_j | j \in [i - n, i - 1]\}$ as the context. We use the contextual n sentences to predict the current sentence via our hierarchical

System	Accuracy
Lin et al. [17]	67.02%
Mihaylov and Frank [20]	72.42%
Schwartz et al. [31]	75.20%
Cai et al. [2]	74.70%
Chaturvedi et al. [6]	77.60%
BOW	73.54%
Embedding	76.45%
CNN	75.38%

Table 1: SCT test results of our proposed supervised models compared with state-of-the-art models.

encoder-decoder. In this paper, the number of context sentences is randomly sampled from $[1, 20]$. The objective for pre-training is to minimize the negative log likelihood of the target sentences.

3 EXPERIMENT

In the experiment, we first show that adopting the development set in SCT for training, is not an appropriate setting to evaluate the context inference ability of a system. Then we compare our model with other unsupervised architecture, including generative and discriminative models, and demonstrate the specific advantage of our model.

3.1 Revisiting Supervised Setting

In this setting, similar with previous methods on SCT that achieve state-of-the-art result, we adopt the development set in SCT, which contains 1871 labeled story (i.e., each story combined with both positive and negative target ending sentence), for training. So in this setting SCT is reduced to a 2-class classification problem. We proposed three simple methods which are purely based on word information as baselines:

- **BOW**: We use bag-of-words as the feature vector for each target sentence, and each word is weighted by its Tf-IDF. We use a simple logistic model for classification.
- **Embedding**: This model is similar with fastText [12]. We use the average word embedding of each sentence as the feature vector. Then a linear layer is applied for classification.
- **CNN**: This model is similar with Embedding, we apply convolutional neural networks on the word embedding, which captures the local information of the input.

It needs to mention that to emphasize the characteristic of the labeled data in SCT, all of the proposed supervised models are merely based on the ending sentence, i.e., we did not take the context information into account. We use 300-d Glove [24] representations as the word embedding. CNN windows size was set to 3. We compare these supervised models with five state-of-the-art models on SCT:

- Lin et al. [17] proposed a method based on heterogeneous knowledge. They adopt sentiment, event relationships, discourse relations etc. as features for classification.
- Mihaylov and Frank [20] proposed two models for classification: 1) features: it use several similarity score as the feature. 2) neural: it use attention based neural networks to modeling the sentence.
- Schwartz et al. [31] use several sophisticated features, such as sentence length, word frequency, word n-grams, character n-grams etc. as features for classification.
- Chaturvedi et al. [6] proposed model based on sophisticated features, such as event relations in FrameNet [1], sentiment trajectory, topical consistency etc.
- Cai et al. [2] proposed a simple neural based attention model to model the sequence.

The result is shown in Table 1.

We can see from the table that although the proposed supervised models are relatively simple, they could achieve a similar result with previous works. Given that these models are merely based on the ending sentence, and did not take the context four sentences into account, this reveals the fact that: *the right prediction could be made by only using the ending sentence information* [1]*, without the need to find clues in the context. So models trained on the labeled development set may not reveal the story comprehension ability.*

On the contrary, in this paper, we proposed an unsupervised generative model to derive the *context-target* probability, which directly modeling the commonsense inference process in a story. In the next several sections, we only use the unlabeled training data, where we could not access to the negative ending sentence. And we evaluate our model in development and test set.

3.2 Common Setup

The SCT training set contains 98167 stories, both dev and test set contains 1871 stories. For the pre-training Bookstory dataset, we remove the target sentences that is too short or too long. For Wikipedia, we use the 2018-06-01 wikidumps[2] and extract only the text passages and ignore lists, tables, and headers. After preprocessing we get more than 90 millions <context, target> pairs.

For our hierarchical encoder-decoder model, we set the word embedding size to 1024, and sentence level LSTM encoder and document level encoder are 4-layers LSTM. We use the byte pair encoding (BPE) to fixed the vocabulary size to 35k. Batch size is set to 256 in the pre-training and 32 in the fine tuning step. We use vanilla dropout [34] on the word embedding layer and variational dropout [10] on the LSTM output layer, with a drop probability 0.1. We use Adadelta [40] with $\rho = 0.999$ to update the parameter. L_1 and L_2 criteria with weight 10^{-5} are added to regulate the parameter. And we adopt a 100k warm-up steps and final 100k training steps during the pre-training period. For all experiments without pre-training, we halve the encoder size to prevent overfitting.

3.3 Baselines

There are several baseline methods proposed in [22] such as using word embedding similarity, sentiment tendency prediction, etc. We compare four of them in this paper: **GenSim**: choose the candidate with its average word embedding closer to the context. **Narrative Chains-AP**: Implements the standard approach to learning chains of narrative events based on Chambers and Jurafsky [3] and chooses the hypothesis whose co-referring entity has the highest average PMI score with the entity's chain in the document. **Narrative Chains-S:**

[1] which is referred as writing style [31].

[2] https://dumps.wikimedia.org/enwiki/

	Gensim	Narrative-Chains-AP	Narrative-Chains-S	DSSM	LSTM	LM	HLSTM w/o att	HLSTM	HLSTM+LR	HLSTM+LR+PT	Supervised SOTA
Dev	0.545	0.472	0.510	0.604	0.621	0.618	0.642	0.647	0.694	**0.775**	0.772
Test	0.539	0.478	0.494	0.585	0.612	0.609	0.631	0.660	0.702	**0.753**	0.765

Table 2: Accuracy of different unsupervised methods in SCT. Our hierarchical LSTM based encoder decoder model is denoted as HLSTM. *w/o att* means the model is not equipped with attention mechanism. LR denotes *loss reweight*, and PT denotes the pre-training. Supervised SOTA is a lexical matching method trained on development set which achieves the best result [33] in SCT.

The same model as above one but is trained on Story Cloze Test. **Deep Structured Semantic Model (DSSM)**: This model is trained to project the context and the fifth sentence into same space [14].

In addition to the above baselines, we also compare our methods with some other unsupervised model that are generatively based on encoder-decoder architecture.

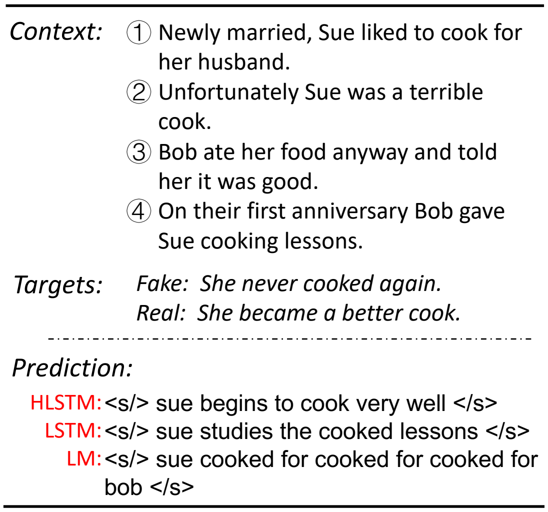
- **Language Model (LM)**: is a model similar to [31], which treats each story as a single sentence and decodes it with an LSTM. Thus the *context* information is modeled on-the-fly by the LSTM.
- **LSTM**: It has the same decoder architecture with the proposed model, but the encoder is a single LSTM model that processes the four context sentences as one sequence.

The result is shown in Table 2[3].

We can see from the table that our proposed hierarchical encoder-decoder model significantly outperforms other models that trained with the unlabeled data. For the Gensim methods, as it only takes the word information into account. However, the inference in SCT is much more difficult which need the more complex semantic composition from words. For the two feature engineering methods based on narrative chains, Mostafazadeh et al. [22] only takes the entity into account, which is not appropriate given that the ending sentences may share the same entities.

Besides, we find that the language model (LM) or sentence level model (LSTM) does not perform comparative result with our hierarchical model. We conjecture that the SCT is a more complicated inference task compared with the previous task such as recognizing textual entailments. The commonsense conveyed by the story sentences is diverse and intricate, so it may not be modeled by simple sequence model. In this paper, we model the context with a hierarchical sequential model, which has better representation capacity and thus achieves better performance.

To better understanding the advantage, we randomly sample the predicted ending of these models in Figure 3. We can see that as The LSTM model did not take the context-architecture into account, its prediction is just coherent with the 4th sentence in context. In the language model, it doesn't discriminate context and target, so during greedy decoding the generated tokens only present the word-by-word coherence. The ending sentence generated by our HLSTM model is most coherent with the context and most similar with the real target sentence. The advantage of HLSTM also accord with previous works such as dialogue systems [32], document summarization [5] who also show hierarchical models are sometimes excel at representing the document level information.



Context: ① Newly married, Sue liked to cook for her husband.
② Unfortunately Sue was a terrible cook.
③ Bob ate her food anyway and told her it was good.
④ On their first anniversary Bob gave Sue cooking lessons.

Targets: *Fake: She never cooked again.*
Real: She became a better cook.

Prediction:
HLSTM:<s/> sue begins to cook very well </s>
LSTM:<s/> sue studies the cooked lessons </s>
LM:<s/> sue cooked for cooked for cooked for bob </s>

Figure 3: Predictions of different models given context.

3.4 Unsupervised Pre-training

We can see in table 2 that the unsupervised pre-training step improves our model significantly. To make a deeper exploitation of the unsupervised pre-training we conduct several experiments:

1) Instead of taking a lot of context sentences, we only use one context sentence to pre-train our model. This is similar with **Skip-though** [16] which also uses LSTM encoder to encode the source sentence and two decoders to predict the previous and following sentence.

2) We do not fine-tune our model on SCT but directly use the pre-trained model for inference.

3) We remove either Wikipedia or BookCorpus to evaluate their contribution during pre-training.

The result of different unsupervised pre-training is shown in Table 3.

We can see that the skip-thought, which only takes one sentence into account, is not competitive for SCT. The intuition behind Skip-Thought is to model the sentence-to-sentence inference process, so it is very suitable to sentence similarity or entailment tasks [16]. However, in SCT, the inference is beyond sentence level so the Skip-Though could not achieve good result. In addition, it is surprising

[3] For the LM, we reimplement the language model introduced in [31], unfortunately we could not obtain the same result. To fairly compare our model with them we report the result of our implementation which has a same software settings with other baselines.

	dev	test
Skip-Thought	0.603	0.589
w/o fine tune	0.692	0.703
w/o Wikipedia	0.752	0.749
w/o BookCorpus	0.702	0.700
pre-training+fine tuning	**0.762**	**0.759**

Table 3: Result of different pre-training strategy. It needs to mention that our our model (w/ fine tune) is not enhanced with loss-reweight training.

that the model without fine-tuning could also achieve a very good result in SCT, which demonstrate the rich semantic information are well modeled by our encoder-decoder, and the knowledge could be transformed to SCT. For the two pre-training resources, as the textual style of Wikipedia is very formal, which is different from SCT. However, the BookCorpus is narrative stories and more similar to SCT, so its influence is more significant.

3.5 Our methods vs. Discriminative methods

The proposed unsupervised models try to learn the *discriminative* pattern without the negative sample. On the contrary, traditional models to deal with textual inference tasks mostly build upon a discriminative architecture [8, 18] in which both positive and negative samples are present in the data. Thus we compare our model with these discriminative models. To make the discriminative classifier available, we proposed three ways to generate the *negative* sentences. **I**: Randomly sample a sentence from the training dataset. **II**: Randomly shuffle the positive sentence. **III**: Randomly generate a sequence from the word vocabulary. It needs to mention that the second and third method may generate the ungrammatical sentence.

In this paper, we proposed three types of deep learning based discriminative models to calculate the score of a <*context-target*> pair:

(1) **HLSTM-D**: we use the HLSTM to get the document representation **c**, and the sentence LSTM encoder to get the target sentence representation **h**. We compare the document with target by their dot value: $score = \sigma(\mathbf{o}^T \cdot \mathbf{h})$ and σ is sigmoid function. We adopt max-margin hinge loss as the training objective:

$$\mathcal{L} = max\{0, M - score_+ + score_-\} \tag{9}$$

where the $score_+$ and $score_-$ are scores for the positive and negative target. M is a pre-defined margin which is set to 0.15.

(2) **AHLSTM-D**: [8] is similar with HLSTM-D, except that the context representation **c** is not average of each context sentence representation $\mathbf{o}_i$ but an attentive weighted sum of them.

(3) **CGANs**: [39] uses a conditional generative adversarial networks [11] to generate the negative targets and then apply a hierarchical discriminator on them.

To fairly compare our model with this discriminative classifiers, we do not apply the loss-reweight or data-enhancement on our HLSTM and only use the original stories. The result is shown in Table 4.

		Test	Dev
Discriminative	HLSTM-D	0.571	0.584
	AHLSTM-D	0.599	0.603
	CGANs	0.609	0.625
Generative	HLSTM	0.694	0.702

Table 4: Result of our proposed models compared with discriminative classifiers. The term 'generative' is not accord to the generative model in machine learning, but a conceptual reference of models that generate text.

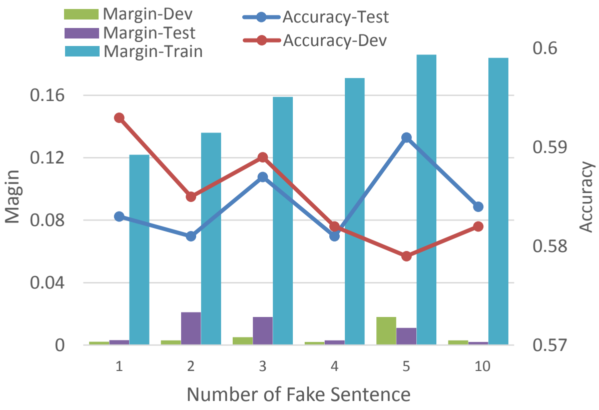


Figure 4: Average accuracy and hinge margin w.r.t. the number of fake examples. Histogram is the margin and solid line is the accuracy.

We can see that although enhanced with attention mechanism, the discriminative model could not outperform the proposed unsupervised methods. This may be attributed to the fact that the randomly generated example cannot provide enough discriminative information for the classifier. To see this more concrete, we generate more and more negative sentences to the discriminative classifier, then given two candidates, we calculate the margin between the larger score and the smaller one based on Equation 9. The result is shown in Figure 4.

We can see that the negative sampling methods for the discriminative classifier are ineffective for inference. In the previous works, when it hard to normalize the probability, such as word embedding [21] or large vocabulary language modeling [15], they usually adopt negative sampling or importance sampling to sample words or entities. However, the sentence space is so huge compared to word space making the randomly sampled negative pattern hard, if not impossible, to be used for probability normalization, which hinders the discriminative generalization of the classifier.

3.6 The Improvement by Loss Reweight

An important innovation in this paper is that we propose a loss reweight strategy for training the encoder-decoder model. In table 2 we show that our model achieves better results when enhanced with the loss-reweight strategy. We use two criteria to measure this improvement: **I**, following traditional machine translation evaluation,

	BLEU-2	BLEU-4	Test	Dev
L-sum	0.3883	0.3382	0.635	0.638
L-mean	0.3719	0.3221	0.632	0.634
LR-G	0.4141	0.3335	0.644	0.638
LR-A	0.3699	0.3292	0.659	0.628
w/o SLLR	0.4101	0.3423	0.673	0.668
w/o DLLR	0.4225	0.3512	0.674	0.682
HLSTM+LR	**0.4413**	**0.3921**	**0.694**	**0.702**

Table 5: The result of different training strategy in terms of the translation quality and SCT accuracy. We set p to 1.2 for LR-G and $\frac{1}{n-1}$ for LR-A.

we use BLEU score as the metric to measure the similarity degree between the generated texts and the ground-truth texts. Specifically, we set n-gram to be 2 (BLEU-2) and 4 (BLEU-4). **II**, The improvement of downstream accuracy in SCT. We design several types of comparison experiment:

(1): Ablation experiments: Remove the sentence level loss reweight (**w/o SLLR**) or document level loss reweight (**w/o DLLR**).

(2): Replace the weighted loss with the summation of each word loss (**L-sum**).

(3): Replace the weighted loss with the average of each word loss (**L-mean**).

(4): The proposed hypothesis is that the loss weight of each word should correlate to the number of words that have been fed to the decoder. So we set each word loss in a sentence linear to its position. Concretely, given a sentence with length n, the loss weight w_i for the ith word is:

(LR-G) Geometric:

$$\begin{aligned} w_i &= \frac{n(p-1)p^i}{p^n-1} \\ s.t. \quad & p > 1 \end{aligned} \tag{10}$$

which means the loss for ith word is p times larger than the loss for $(i-1)$th.

(LR-A) Arithmetic:

$$\begin{aligned} w_i &= 1 - p(\frac{n-1+2i}{2}) \\ s.t. \quad & 0 < p < \frac{2}{n-1} \end{aligned} \tag{11}$$

The result is shown in Table 5.

We can see from the table that when training the decoder with loss reweight strategy, the performance (both in terms of the generated sentence quality and inference accuracy) could be improved. Best results are achieved when we employ the self-determined loss reweight strategy. In this manner, the loss is self-adjusted during the different training period and among different training instances. As the training procedure is tuned by the model itself, so it would be better adjust the learning process and achieves a better result.

4 RELATED WORK

Machine comprehension is a recently proposed natural language understanding task which aims at teaching a machine to understand the text and accomplish question answering or textual inference problem. Since the MCTest [29] was proposed, many researchers have been focused on this task. Hermann et al. [13] proposed a large cloze style CNN/Daily Mail dataset in which the target is to generate the word in a statement slot given the context. However, this dataset is derived semi-automatically from the newspaper and the target words are limited to nouns, which confines the inference ability required to answer the questions [7]. SQuAD [28], NewsQA [38] and MARCO [23] are recently released MC datasets. Most of the questions in these datasets are limited to syntactic variation or lexical variation [35]. In this paper, we are focused on SCT, which evaluates the deeper semantic inference ability of a system. The baseline models proposed in [22] contain not only feature engineering systems but also deep learning models, nonetheless, the performance is still poor compared with human. Schwartz et al. [31] proposed some supervised methods based on the writing style of the annotator, which achieves a good result on SCT. Mihaylov and Frank [20] proposed a lexical matching method on this data, which compares the two ending candidates by n-gram overlap. Lin et al. [17] proposed to employing external knowledge, such as event relations, to deal with SCT. Chaturvedi et al. [6] also proposed a model based on linguistic features and model them with a hidden coherence model. Cai et al. [2] proposed a simple neural based attention model with LSTM. Despite substantial improvement over baselines, these methods are based on the labeled development data, which did not fully take the unlabeled training set into account.

Script Learning is a canonical representative of traditional textual inference methods. It processes the temporally ordered sequences of symbolically structured events and tries to predict future events. Previous methods are non-probabilistic and brittle which pose serious problems for automated learning. In recent years, there has been a growing body of research into statistical script learning, which enables statistical inference of implicit events from the text [26, 30]. Chambers and Jurafsky [3, 4] describe some simple event co-occurrence based systems which infer (verb dependency) pairs related to a particular discourse entity. However, these methods rely heavily on dependency parser and co-reference tools to transform the document into event chains. Which brings noise to SCT.

Unsupervised Pre-training Unsupervised pre-training has been widely adopted in the deep learning era. In computer visions, many methods first pre-train their models on Imagenet and then fine-tuned on the task at hand. In NLP community, instead of just optimizing the models from random initialization, some parameters are initialized by an unsupervised application in a large amount of data. For example, word2vec [21] or Glove [24] are two representative methods to initialize the word embedding. However, they only take the word information into account. ELMO [25] and GPT [27] try to initialize the sentence embedding by a language model. Skip-thoughts [16] is another sentence embedding initialization methods which are similar to our proposed methods, but they only take the nearby sentence into account which may not capture the long-term dependencies between sentences. Very recently, Devlin et al. [9] proposed a self-attention model to initialize the sentence embedding by a mask language model, which obtains significant improvements in many NLP applications. But their pre-training is focus on sentence level initialization and must be fine-tuned during inference. In contrast, our initialization is exactly same with the inference process, and the hierarchical architecture enable our model to capture structure information in the text.

5 CONCLUSION

In this paper, to deal with story comprehension application SCT, unlike most previous works which utilize the small development set for training. We directly modeling the unlabeled stories with two hierarchical encoder-decoder. We develop a self-determined loss reweight training strategy to optimize the decoder. We also adopt a large amount of unlabeled data to pre-training our model and achieve comparative result with supervised models. We demonstrate the advantage of our proposed model compared with other unsupervised generative and discriminative model. In addition, the loss-reweight training strategy proposed in this paper could strengthen the decoding quality of the encoder-decoder model.

REFERENCES

[1] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet Project. In *COLING-ACL*.

[2] Zheng Cai, Lifu Tu, and Kevin Gimpel. 2017. Pay Attention to the Ending: Strong Neural Baselines for the ROC Story Cloze Task. In *ACL*.

[3] Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised Learning of Narrative Event Chains.. In *ACL*, Vol. 94305. Citeseer, 789–797.

[4] Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *ACL*. Association for Computational Linguistics, 602–610.

[5] Ming-Wei Chang, Kristina Toutanova, Kenton Lee, and Jacob Devlin. 2019. Language Model Pre-training for Hierarchical Document Representations. *arXiv preprint arXiv:1901.09128* (2019).

[6] Snigdha Chaturvedi, Haoruo Peng, and Dan Roth. 2017. Story Comprehension for Predicting What Happens Next. In *EMNLP*.

[7] Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task. In *Association for Computational Linguistics (ACL)*.

[8] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proc. ACL*.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[10] Yarin Gal and Zoubin Ghahramani. 2016. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. In *NIPS*.

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *NIPS*.

[12] Edouard Grave, Tomas Mikolov, Armand Joulin, and Piotr Bojanowski. 2017. Bag of Tricks for Efficient Text Classification. In *EACL*.

[13] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*. 1684–1692.

[14] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd CIKM*. ACM, 2333–2338.

[15] Rafal Józefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the Limits of Language Modeling. *CoRR* abs/1602.02410 (2016).

[16] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*. 3294–3302.

[17] Hongyu Lin, Le Sun, and Xianpei Han. 2017. Reasoning with Heterogeneous Knowledge for Commonsense Machine Comprehension. In *EMNLP*. 2022–2033. http://aclweb.org/anthology/D17-1215

[18] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130* (2017).

[19] Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *EMNLP*.

[20] Todor Mihaylov and Anette Frank. 2017. Story Cloze Ending Selection Baselines and Data Examination. *arXiv preprint arXiv:1703.04330* (2017).

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*. 3111–3119.

[22] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. *Proceedings of NAACL HLT* (2016).

[23] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv preprint arXiv:1611.09268* (2016).

[24] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*.

[25] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018).

[26] Karl Pichotta and Raymond J Mooney. 2016. Learning Statistical Scripts with LSTM Recurrent Neural Networks.. In *AAAI*. 2800–2806.

[27] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf* (2018).

[28] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *EMNLP*.

[29] Matthew Richardson, Christopher JC Burges, and Erin Renshaw. 2013. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text.. In *EMNLP*, Vol. 1. 2.

[30] Rachel Rudinger, Pushpendre Rastogi, Francis Ferraro, and Benjamin Van Durme. 2015. Script induction as language modeling. In *EMNLP*. 1681–1686.

[31] Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A Smith. 2017. The Effect of Different Writing Tasks on Linguistic Style: A Case Study of the ROC Story Cloze Task. *arXiv preprint arXiv:1702.01841* (2017).

[32] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[33] Siddarth Srinivasan, Richa Arora, and Mark O. Riedl. 2018. A Simple and Effective Approach to the Story Cloze Test. In *NAACL-HLT*.

[34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. [n.d.]. Dropout: A simple way to prevent neural networks from overfitting. *JMLR* ([n. d.]).

[35] Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What Makes Reading Comprehension Questions Easier? *EMNLP* (2018).

[36] Saku Sugawara, Yusuke Kido, Hikaru Yokono, and Akiko Aizawa. 2017. Evaluation Metrics for Machine Reading Comprehension: Prerequisite Skills and Readability. In *ACL*.

[37] Simon Suster and Walter Daelemans. 2018. CliCR: A Dataset of Clinical Case Reports for Machine Reading Comprehension. *CoRR* abs/1803.09720 (2018).

[38] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. NewsQA: A Machine Comprehension Dataset. *arXiv preprint arXiv:1611.09830* (2016).

[39] Bingning Wang, Kang Liu, and Jun Zhao. 2017. Conditional Generative Adversarial Networks for Commonsense Machine Comprehension. In *IJCAI*.

[40] Matthew D Zeiler. 2012. ADADELTA: an adaptive learning rate method. *arXiv preprint* (2012).

[41] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *The IEEE International Conference on Computer Vision (ICCV)*.