Enhanced Story Representation by ConceptNet for Predicting Story Endings

Shanshan Huang
huangss_33@sjtu.edu.cn
Shanghai Jiao Tong University

Kenny Q. Zhu*
kzhu@cs.sjtu.edu.cn
Shanghai Jiao Tong University

Qianzi Liao
liaoqz@sjtu.edu.cn
Shanghai Jiao Tong University

Libin Shen
libin@leyantech.com
Leyan Tech

Yinggong Zhao
ygzhao@leyantech.com
Leyan Tech

Abstract

Predicting endings for narrative stories is a grand challenge for machine commonsense reasoning. The task requires accurate representation of the story semantics and structured logic knowledge. Pre-trained language models, such as BERT, made progress recently in this task by exploiting spurious statistical patterns in the test dataset, instead of "understanding" the stories per se. In this paper, we propose to improve the representation of stories by first simplifying the sentences to some key concepts and second modeling the latent relationship between the key ideas within the story. Such enhanced sentence representation, when used with pre-trained language models, makes substantial gains in prediction accuracy on the popular Story Cloze Test without utilizing the biased validation data.

CCS Concepts: • **Computing methodologies → Reasoning about belief and knowledge**.

Keywords: commonsense reasoning; story comprehension; commonsense knowledge

ACM Reference Format:
Shanshan Huang, Kenny Q. Zhu, Qianzi Liao, Libin Shen, and Yinggong Zhao. 2020. Enhanced Story Representation by ConceptNet for Predicting Story Endings. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20), October 19–23, 2020, Virtual Event, Ireland.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3340531.3417466

*Corresponding Author.

CIKM '20, October 19–23, 2020, Virtual Event, Ireland

ACM ISBN 978-1-4503-6859-9/20/10…$15.00
https://doi.org/10.1145/3340531.3417466

1 Introduction

Predicting "what happens next" in narrative stories is an important but challenging task of commonsense reasoning in AI. Story comprehension was first studied in the context of planning and goal searching [8], which was one of the most important problems in AI. The task evolved to predicting what is expected to happen next in stories. Much work has been evaluated on a standard dataset called Story Cloze Test (SCT) [9]. SCT asks for the correct ending of a four-sentence story context from two alternatives, as shown in Figure 1(a).

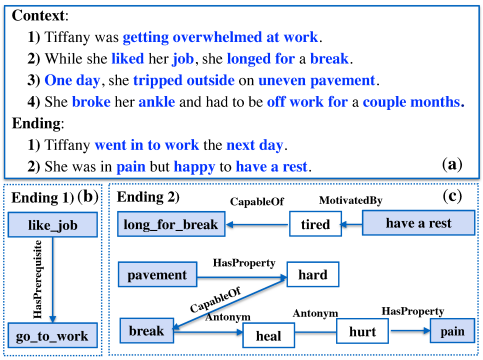

Figure 1. An example from the story-cloze task. In (b) and (c), the words in the blue boxes are concepts from the story; the words in the white boxes are concepts not from the story but serve as bridging nodes.

Previous works suggest that structured commonsense knowledge [6, 13] may enhance story understanding. For example, from Figure 1 (b) and (c), the structured knowledge can help reason story endings with logic relations between tokens. Meanwhile, it is easy to find that one could arrive at the correct ending 2) by looking at only some of the key words (highlighted in blue) which are more informative for inference, instead of consuming all the words. In fact, the other un-highlighted words are not only uninformative, but may even confuse the downstream classifier with ambiguous semantics. For example, the name Tiffany is often associated with jewelries, thus the introduction of this meaning into the story context does more harm than good.

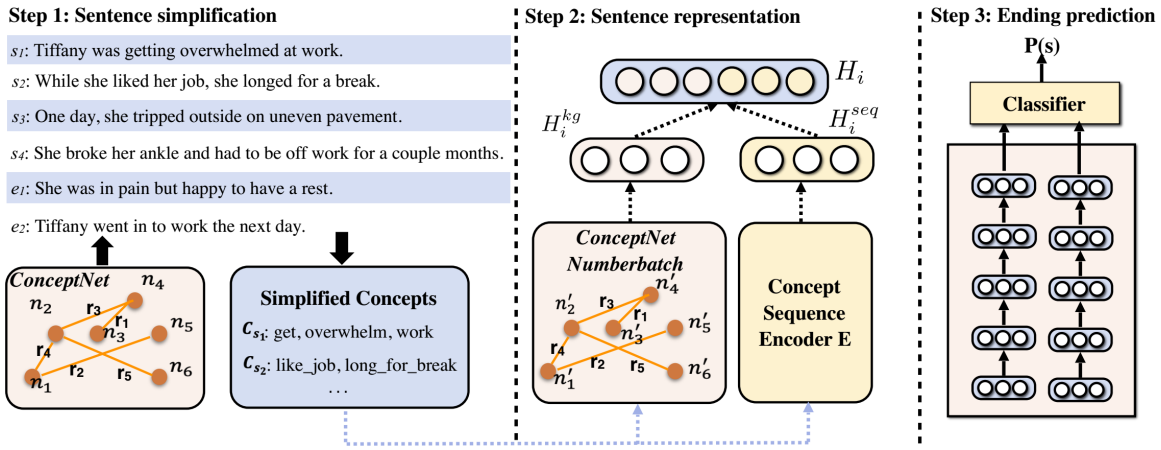

Figure 2. Framework overview: our framework can be divided into three steps: sentence simplification, sentence representation and ending prediction. $n_1, ...n_6$ are the examples of token nodes in ConceptNet, $r_1, ...r_6$ are the commonsense relations between the nodes. $n_1^{'}, ...n_6^{'}$ are the corresponding vector from Numberbatch which are trained on ConceptNet.

Inspired by the above observation, we improve the story representation by using commonsense knowledge from two aspects. First, we simplify sentences by extracting a sequence of concepts from ConceptNet [15], a community curated open-domain knowledge graph covering much of the knowledge required for commonsense reasoning and obtain the *intra-sentence* concept representation. Second, we incorporate structured commonsense knowledge from ConceptNet by including the pre-trained concept embeddings from ConceptNet knowledge graph to story sentence representation. For example, in Figure 1(c), "long for break" is related to "have a rest" through *CapableOf* and *MotivatedBy* relation edges. These edges help us "connect the dots" within the story and allow us to make more meaningful deduction along the story.

In summary, this paper makes the following contributions:

- We simplify the stories by streamlining sentences to a few key concepts, which eliminates unwanted variance in the text, and achieve better results than using the original sentences (see Section 2.1).
- We combine sequential and structured representation for the key concepts in a sentence as the sentence representation and get better performance (see Section 2.2).
- Our approach, when combined with the suitable language model, beats the recent state-of-the-art methods by substantial margins using the corrected, unbiased training data (see Section 3.2 and Section 3.3).

2 Approach

Given the story context $\mathbf{s} = (s_1, s_2, ..., s_L)$ of L sentences, the goal of the problem is to predict the correct ending from two candidate ending sentences e_1 and e_2. We propose to refine and extend story sentence representations through incorporating commonsense knowledge from ConceptNet for story ending prediction. The architecture is shown in Figure 2, which consists of the following steps.

2.1 Sentence Simplification

Our goal is to extract a sequence of concepts C_s from an input sentence $s = \{w_1, ..., w_N\}$ with N words. C_s contains only the key concepts in s. [1] We choose to use ConceptNet [15] as the source of these concepts because of its comprehensive coverage of commonsense knowledge.

The concepts in ConceptNet are expressed in terms of short phrases, commonly made up of one or two words such as "break ankle". While these phrases are meaningful and understandable to human beings, they may not find exact match in the input sentences, simply because people don't say "break ankle" in normal text but "break her ankle" instead. To remedy this problem, we develop a fuzzy match method, that allows one additional words to be inserted to a concept from ConceptNet before exact matching in the input sentence.

Another technical issue is that extracted concepts may overlap with each other in the input sentence. For example, from the sentence "She hope it would come back for more later", we can extract the following concepts: "hope", "**come back**", "**come for**", and "more". "Come back" and "come for" are both meaningful, and we keep them in C_s. Afterwards, we remove duplicate concepts from C_s, if it's contained by other concepts in the sequence. For example, we match "come" and "come back" in sentence. "Come" will be deleted because it is the sub-sequence of the words in "come back".

2.2 Sentence Representation

After sentence simplification, the original sentence s is transformed into a sequence of concepts C_s with the same order

[1] The input sentences are tokenized and lemmatized using Stanford CoreNLP [7]

in s. We present the methods to encode the concept sequence and the concept graph respectively.

Concept Sequence Encoding: After simplification process, the concept sequence of a sentence is encoded into vector representation using a sequential encoder E. Concept sequence C_s is converted into a flatten word sequence s', which is the concatenation of the words of all the concepts in C_s. Compared with the original sentence, s' is a simplified word sequence where commonsense-irrelevant information has been discarded. In the above case, the simplified sequence will be "hope come back come for more".

Then simplified sequence s' is fed into a sequential encoder, which maps the input s' into a sequence of contextual embeddings H^{seq}:

$$H^{seq} = E\left(s'\right) \tag{1}$$

Concept Encoding: Besides the flattened concept sequence in the simplified sentence, the relation between concepts is also important for predicting the story ending. We take advantage of pre-trained concept embedding from Numberbatch[2] which is the pre-trained on ConceptNet knowledge graph containing more than 2,000,000 popular concepts. Given the sequence of concepts extracted from a sentence, C_s, we define the structured knowledge representation H^{kg} as the sum of each concept:

$$H^{kg} = \sum_{c \in C_s} Numberbatch(c), \tag{2}$$

where $Numberbatch(c)$ denotes the concept vector of concept c. If the concept is not in Numberbatch, we approximate its concept vector by averaging the vectors of all its constituent words which can be found in Numberbatch.

Finally, the complete representation of the sentence s is defined as the concatenation of two components: $H_s = [H_s^{seq}; H_s^{kg}]$. The representation will be fed into a fully connect layer for choice classification.

3 Evaluation

3.1 Baselines

The baseline models can mainly be divided into supervised and unsupervised:

Unsupervised: DSSM [9] and GMSA [3] calculate semantic similarity between a pair of strings by representing them in a continuous semantic space.

Supervised: CGAN [18] generates negative endings as training the discriminator. SKBC [12] and SIMP [16] uses Skip-thought [4] with GRU-GRU structure to produce generic sentence representations. BERT [2] and TransBERT [6] apply the bidirectional training of Transformer [17] compared to unidirectional Transformer for GPT [11] and ISCK [1]. Except for the pre-training representation models, FES-JOINT [10] and SeqMANN [5] make fully use of various semantic features, like sentiment, to get better results.

We choose to apply our methods on 3 typical models: DSSM, GPT and BERT.

3.2 Dataset

Model	Endings of SCT(V) (%)	Endings of SCT(R) (%)
SIMP	72.60	59.86
SKBC	72.76	58.18
GPT	77.77	57.93
$\text{TransBERT}_{\text{BASE}}$	79.0	54.52
$\text{TransBERT}_{\text{LARGE}}$	75.84	54.30
Human	62.40	62.40

Table 1. SCT test accuracies of SOTA models trained from endings only in SCT(V) vs. endings only in SCT(R).

The SCT dataset comes with 101,903 5-sentence stories (first 4 as context and last as ending). Human authors were asked to write negative endings for 3744 of these stories to create cloze test instances. These 3744 instances were then split into validation set (SCT(V)) and test set (SCT(T)). The remaining 98,159 stories are called raw stories. Previous work indicated that human-authorship bias [14] exists in SCT datasets, especially when the validation set of SCT is used for training. In fact, in a "stripped-down" version of the SCT task, where one is supposed to choose between two alternative story endings, without given the context, SOTA models all performed much better than human, after training from the endings-only data of the SCT validation set. This shows that the models are not really capable of human-like reasoning, but merely pick up cues from the endings in the validation set (see SCT(V) column of Table 1).

We thus follow Roemmele et al.'s [12] to construct a new training set called SCT(R) by Random and Backward sampling of negative endings for 98,159 raw stories. With the new training data, the same SOTA models performs reasonably worse than human in the "ending-only" test, as shown in SCT(R) column of Table 1. Therefore in the remaining experiments, we will use SCT(R) as the training data for all competing algorithms in both SCT and $\text{SCT}_v1.5$ [3] tests.

3.3 End-to-end Results

We first show the end-to-end results of the three baselines with simplification and concept encoding methods. Then we evaluate other models trained with new dataset on SCT. In Table 2, SKBC and DSSM achieve significant 3.43% and 4.75% improvement with simplification. $\text{BERT}_{\text{BASE}}$ gains 0.8% increase with simplification and 2.89% with concept encoding method (both compared with original model). BERT may

[2] https://github.com/commonsense/conceptnet-numberbatch

[3] $\text{SCT}_v1.5$ was previously released on https://competitions.codalab.org/competitions/15333#participate-submit_results, with the goal of fixing some of the bias, but was later found to have other problems and was subsequently closed for access.

Model	Test	Original (%)	Simp(%)	CE(%)	Simp+CE (%)
DSSM		54.04	**58.79**	54.0	**58.2**
SKBC	SCT	64.70	**68.13**	**65.12**	**69.7**
$\text{BERT}_{\text{BASE(ours)}}$		56.54	**57.34**	**59.43**	**60.24**
DSSM		54.30	**57.83**	**54.35**	**58.53**
SKBC	$\text{SCT}_{\upsilon}1.5$	64.56	**67.30**	**65.45**	**67.97**
$\text{BERT}_{\text{BASE(ours)}}$		56.88	**58.02**	**59.79**	**60.97**

Table 2. End-to-end accuracy on SCT and $\text{SCT}_{\upsilon}1.5$ test sets. Original=baseline, Simp=simplification method, CE=concept encoding method

Model	Acc (%))	Model	Acc (%))
DSSM	54.04	SKBC	64.70
GMSA	61.20	CGAN	60.90
SeqMANN	59.74	$\text{BERT}_{\text{BASE(ours)}}$	56.54
SIMP	61.09	$\text{BERT}_{\text{BASE}}$	61.46
FES-LM	61.60	$\text{BERT}_{\text{LARGE}}$	64.67
ISCK	62.21	$\text{TransBERT}_{\text{BASE}}$	61.46
GPT	63.46	$\text{TransBERT}_{\text{LARGE}}$	61.89
SKBC+Simp+CE(ours)	**69.7**	Human	100

Table 3. Results of story ending prediction on SCT [4]

learn the informative weight from pre-training with Transformer unit. Our simplification can even help with reducing the weight of less informative words for BERT. DSSM+CE performs worse than DSSM mainly because DSSM is a bag-of-words model and inevitably loses the order information. We can also get the same conclusion from the results testing on $\text{SCT}_{\upsilon}1.5$ that simplification and graph embedding can benefit ending prediction.

Table 3 shows the results of other previous research on our new training data and test on SCT test set. $\text{BERT}_{\text{BASE(ours)}}$ retrains the language model of $\text{BERT}_{\text{BASE}}$ with BookCorpus. Our $\text{BERT}_{\text{BASE}}$ performs worse than the basic version because we retrain the language model with less unsupervised data. Though larger corpus, such as Wikipedia, can lead to a better result, we only expect to show the effective improvement of our simplification and concept encoding methods. SKBC with our methods achieves 69.7% accuracy, which is the best among our experiments. It performs better than any other commonly-used models we tested. Notice that our experiments are not meant to demonstrate the superiority of a particular algorithm but to show that the proposed story representation methods (i.e., simplification and concept encoding) work for a variety of models. Human performance is 100% and can be viewed as an upperbound [9]. All results are averaged from 5 independent runs.

4 Conclusion

Our approach well integrated the ideas of main information extraction and structured knowledge incorporation and get better performance with automatically generated unbiased dataset. From the results we can find that predicting story ending is still a challenging task in artificial intelligence with little high quality data. We consider to generate higher quality datasets for training as future work.

[4]Some scores are great lower than that of the published models because they are trained with validation data which contains spurious features and proved to be unable for training in Section 3.2.

5 Acknowledgments

This work was partially supported by NSFC grant 91646205, SJTU Medicine-Engineering Cross-disciplinary Research Scheme and SJTU-Leyan Joint Research Scheme.

References

[1] Jiaao Chen, Jianshu Chen, and Zhou Yu. 2018. Incorporating Structured Commonsense Knowledge in Story Completion. *CoRR* (2018).

[2] 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).

[3] Jian Guan, Yansen Wang, and Minlie Huang. 2018. Story ending generation with incremental encoding and commonsense knowledge. *arXiv preprint arXiv:1808.10113* (2018).

[4] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *NIPS*.

[5] Qian Li, Ziwei Li, Jin-Mao Wei, Yanhui Gu, Adam Jatowt, and Zhenglu Yang. 2018. A Multi-Attention based Neural Network with External Knowledge for Story Ending Predicting Task. In *COLING*.

[6] Zhongyang Li, Xiao Ding, and Ting Liu. 2019. Story Ending Prediction by Transferable BERT. *arXiv preprint arXiv:1905.07504* (2019).

[7] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL*.

[8] James R Meehan. 1977. TALE-SPIN, An Interactive Program that Writes Stories.. In *IJCAI*.

[9] 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. In *NAACL*.

[10] Haoruo Peng, Snigdha Chaturvedi, and Dan Roth. 2017. A joint model for semantic sequences: Frames, entities, sentiments. In *CoNLL*.

[11] Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. 2018. *Improving language understanding with unsupervised learning*. Technical Report. Technical report, OpenAI.

[12] Melissa Roemmele, Sosuke Kobayashi, Naoya Inoue, and Andrew Gordon. 2017. An rnn-based binary classifier for the story cloze test. In *LSDSem*.

[13] Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *AAAI*.

[14] Rishi Sharma, James Allen, Omid Bakhshandeh, and Nasrin Mostafazadeh. 2018. Tackling the Story Ending Biases in The Story Cloze Test. In *ACL*.

[15] Robert Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*.

[16] Siddarth Srinivasan, Richa Arora, and Mark Riedl. 2018. A Simple and Effective Approach to the Story Cloze Test. In *ACL*.

[17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

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