Composing a Picture Book by Automatic Story Understanding and Visualization

Xiaoyu Qi[1], Ruihua Song[1], Chunting Wang[2] , Jin Zhou[2], Tetsuya Sakai[3]
[1]Microsoft, Beijing, China
[2]Beijing Film Academy, Beijing, China
[3]Waseda University, Tokyo, Japan
{xiaqi, rsong}@microsoft.com, 04172154@mail.bfa.edu.cn
whitezj@vip.sina.com, tetsuyasakai@acm.org

Abstract

Pictures can enrich storytelling experiences. We propose a framework that can automatically compose a picture book by understanding story text and visualizing it with painting elements, i.e., characters and backgrounds. For story understanding, we extract key information from a story on both sentence level and paragraph level, including characters, scenes and actions. These concepts are organized and visualized in a way that depicts the development of a story. We collect a set of Chinese stories for children and apply our approach to compose pictures for stories. Extensive experiments are conducted towards story event extraction for visualization to demonstrate the effectiveness of our method.

1 Introduction

A story is an ordered sequence of steps, each of which can contain words, images, visualizations, video, or any combination thereof (Kosara and Mackinlay, 2013). There exist vast amounts of story materials on the Internet, while few of them include visual data. Among the few presented to audience, some include illustrations to make the stories more vivid; others are converted to video forms such as cartoons and films, of which the production consumes a lot of time and human efforts. Although visualized stories are difficult to generate, they are more comprehensible, memorable and attractive. Thus, automatic story understanding and visualization has a broad application prospect in storytelling.

As an initial study, we aim to analyze events of a story and visualize them by combining painting elements, i.e., characters and backgrounds. Story understanding has been a challenging task in Natural Language Processing area for a long time (Charniak, 1972). In order to understand a story, we need to tackle the problem of event extraction in a story. A story usually consists of several plots, where characters appear and make actions. We define event keywords of a story as: scene (where), character (who, to whom) and action (what). We extract events from story on both sentence level and paragraph level, so as to make use of the information in each sentence and the context of the full story.

As for story visualization, the most challenging problem is stage directing. We need to organize the events following certain spatial distribution rules. Although literary devices might be used e.g. flashbacks, the order in a story plot roughly corresponds with time (Kosara and Mackinlay, 2013). We arrange the extracted events in a screen along the story timeline. Positions of elements on the screen are determined according to both current and past events. Finally, with audio track added, simple animations could be generated. These simple animations are like storyboards, in which each image represents a major event that correspond to a sentence or a group of consecutive sentences in the story text.

Regarding storytelling, we need to first know our audiences, assess their level of domain knowledge and familiarity with visualization conventions (Ma et al., 2012). In this paper, our target is to understand and visualize Chinese stories for children. We collect children's stories from the Internet. (The sources are described in Section 7.1.) Then, we extract events and prepare visualization materials and style for children. The framework we proposed, however, has wide extensibility, since it does not depend on domain specific knowledge. It could serve as an automatic picture book composition solution to other fields and target audience.

Our contributions are threefold. 1) We propose an end-to-end framework to automatically generate a sequence of pictures that represent major

Proceedings of the Second Storytelling Workshop, pages 1–10
Florence, Italy, August 1, 2019.

events in a story text. 2) New formulation of story event extraction from sentence level to paragraph level to align the events in a temporal order. 3) We propose using a neural encoder-decoder model to extract story events and present empirical results with significant improvements over the baseline.

The paper is organized as follows: In Section 2 we introduce related work. Then we formulate the problem and overview our proposed solution in Section 3. Details of different modules are provided in Section 4, 5 and 6. We describe our data and experiments in Section 7. In Section 8 we make conclusion and present our future work.

2 Related Work

2.1 Story Event Extraction

Event extraction is to automatically identify events from text about what happened, when, where, to whom, and why (Zhou et al., 2014). Previous work on event extraction mainly focuses on sentence-level event extraction driven by data or knowledge.

Data-driven event extraction methods rely on quantitative methods to discover relations (Hogenboom et al., 2011). Term frequency-inverse document frequency (TF-IDF) (Salton and McGill, 1986) and clustering (Tanev et al., 2008) are widely used. Okamoto et al. (2009) use hierarchical clustering to extract local events. Liu et al. (2008) employ weighted undirected bipartite graphs and clustering methods to extract events from news. Lei et al. (2005) propose using support vector machines for news event extraction.

Knowledge-driven approaches take advantages of domain knowledge, using lexical and syntactical parsers to extract target information. McClosky et al. (2011) convert text to a dependency tree and use dependency parsing to solve the problem. Aone et al. (2009) and Nishihara et al. (2009) focus on designed patterns to parse text. Zhou et al. (2014) propose a Bayesian model to extract structured representation of events from Twitter in an unsupervised way. Different frameworks are designed for specific domains, such as the work in (Yakushiji et al., 2000), (Cohen et al., 2009) and (Li et al., 2002)). Although there is less demand of training data for knowledge-driven approaches, knowledge acquisition and pattern design remain difficult.

In order to deal with the disadvantages of both methods, researchers work on combining them. At the training stages of data-driven methods, initial bootstrapppings with dependency parser (Lee et al., 2003) and clustering techniques (Piskorski et al., 2007) are used for better semantic understanding. Chun et al. (2004) combine lexicon syntactic parser and term co-occurrences to extract biomedical events while Jungermann et al. (2008) combine a parser with undirected graphs. The only trial of neural network on this task is the work in (Tozzo et al., 2018), where they employ RNN with dependency parser as training initialization.

We propose a hybrid encoder-decoder approach for story event extraction to avoid human-knowledge requirement and better utilize the neural network. Moreover, previous work focus on sentence-level event extraction, which has a gap to apply to full story visualization due to the loss of event continuity. Thus, we extend event extraction to paragraph level so that it is possible to visualize a story coherently in a time sequence.

2.2 Story Visualization

Previous work mainly focuses on narrative visualization (Segel and Heer, 2010), where the visualization intention is deeper understanding of the data and the logic inside. Valls et al. (2017) extract story graphs a formalism that captures the events (e.g., characters, locations) and their interactions in a story. Zitnick et al. (2013) and Zeng et al. (2009) interpret sentences and visualize scenes. There also exists visual storytelling task (Huang et al., 2016).

The most relevant work to ours is that of Shimazu et al. (1988), where they outlined a story driven animation system and presented story understanding mechanism for creating animations. They mainly targeted on interpretations of three kinds of actions: action causality check, action continuity beyond a sentence and hidden actions between neighbouring sentences. The key solution was a Truth Maintenance System proposed in (Doyle, 1979), which relies on pre-defined constrains from human knowledge. Understanding a story with a TMS system would cost a lot of manual efforts. In light of this, we propose an approach to story understanding that automatically learns from labelled story data.

Different from previous work, we propose new story visualization techniques, including temporal and spatial arrangement for screen view. Our

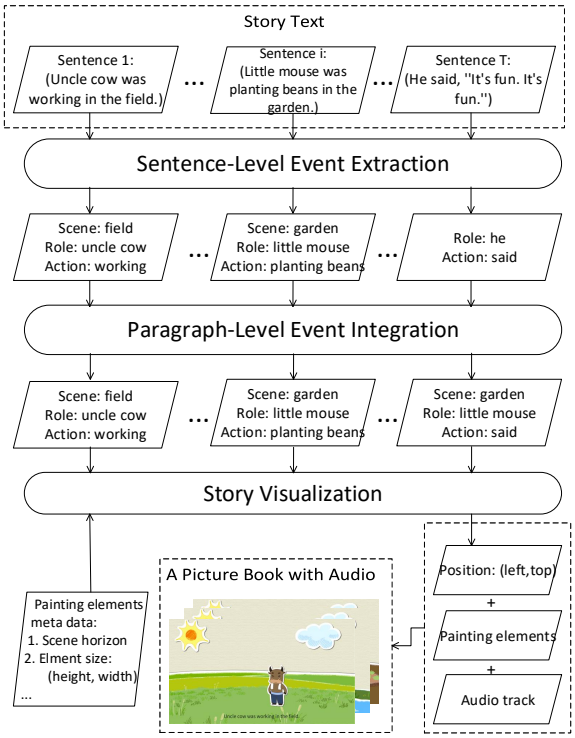


Figure 1: A flowchart for story understanding and visualization

framework generates story animations automatically from end to end. Moreover, it is based on event extraction on both sentence level and paragraph level.

3 Problem Formulation and System Overview

We formulate the problem as follows: the input is a story that contains m paragraphs and each paragraph p contains n sentences, which are composed of several words. The output is a series of images that correspond to the story. An image I is composed by some prepared painting elements (30 scenes, such as sea, and 600 characters, such as fox and rabbit, with different actions in our experiment). As it is costly to prepare painting elements, given a number of stories to visualize, we hope that the fewer elements are prepared the better.

We show the flowchart of our proposed solution in Figure 1. Given a story text, we first split it into a sequence of narrative sentences. Story event extraction is conducted within each sentence independently. Then events are integrated on paragraph level and fed into the visualization stage, where they are distributed temporally and spatially on the screen. The visualization part determines what to be shown on the screen, when and where they should be arranged. Finally, painting elements are displayed on the screen and audio track is added to make a picture book with audio.

4 Sentence-Level Event Extraction

We start from event extraction on sentence-level. Given a sentence $s = (x_1, x_2, ..., x_T)$ of length T, we intend to get a label sequence $y = (y_1, y_2, ..., y_T)$, where $y_i \in scene, character, action, others, i \in [1, T]$. We propose using a neural encoder-decoder model to extract events from a story.

4.1 BiLSTM-CRF

BiLSTM-CRF is the state-of-the-art method to solve the sequence labeling problem. Thus we apply this model to extract events in sentence level.

We can encode the story sentence with a Bidirectional LSTM (Graves and Schmidhuber, 2005), which processes each training sequence forwards and backwards. A Conditional Random Fields (CRF) (Ratinov and Roth, 2009) layer is used as the decoder to overcome label-bias problem .

Given a sentence $s = (x_1, x_2, ..., x_T)$ of length T, we annotate each word and get a ground-truth label sequence $l = (l_1, l_2, ..., l_T)$. Every word $x_i (i \in [1, T])$ is converted into a real-valued vector $e_i (i \in [1, T])$ with a word-embedding dictionary pre-trained from Wikipedia Chinese corpus. Then the sentence is represented as $E = (e_1, e_2, ..., e_T)$, where each e_i is padded to a fixed-length. We set the embedding length to 100 in our experiment. The embedded sentence vector E is fed into a BiLSTM neural network. The hidden state h_i of the network is calculated in the same way as in (Graves and Schmidhuber, 2005).

Different from standard LSTM, Bidirectional LSTM introduces a second hidden layer that processes data flow in the opposite direction. Therefore, it is able to extract information from both the previous and latter knowledge. Each final hidden state is the concatenation of the forward and backward hidden states:

$$h_t = [\overleftarrow{h_t}; \overrightarrow{h_t}] \quad (1)$$

Instead of adding a softmax classification layer after the hidden states, we employ CRF (Ratinov and Roth, 2009) to take the label correlations into consideration. The hidden layer $h = (h_1, h_2, ..., h_T)$ is fed into the CRF layer. We

intend to get the predicted label sequence $y = (y_1, y_2, ..., y_T)$. The conditional probability is defined as:

$$f(y,h) = \sum_{i=1}^{T} W_{y_i}^T h_i + \sum_{i=0}^{T} A_{y_i,y_{i+1}}$$
$$P(y|h) = \frac{\exp(f(y,h))}{\sum_{y'} \exp(f(y',h))} \quad (2)$$

where T is the length of the output sequence. $W_{y_i}^T$ is weight matrix. $A_{y_i,y_{i+1}}$ represents the transitioning score from label y_i to label y_{i+1}. And $y^{'}$ stands for any possible output label sequence. Our training objective is minimizing the negative log likelihood of $P(y|h)$.

4.2 Model Variants

Recently, a new pre-trained model BERT obtains new state-of-the-art results on a variety of natural language processing tasks (Devlin et al., 2018). We apply this model to our story event extraction. We input a sentence to the BERT base model released by Devlin et al. The last layer of BERT serves as word embedding and input of the BiLSTM model. The other parts of the model remain the same for comparison. We refer to this variant as BERT-BiLSTM-CRF.

We also experiment with IDCNN model (Strubell et al., 2017) and fix the parameter setting for comparison. IDCNN model leverages convolutional neural network instead of recurrent one to accelerate the training process.

5 Paragraph-Level Event Integration

When generating a story animation, we need to take consideration of the full paragraph, so that the events could be continuous in temporary order. (A story might consists of one or multiple paragraphs.) In this part, we integrate sentence-level story events to paragraph-level ones. Given a story paragraph $p = (s_1, s_2, ..., s_n)$ of length n, where sentence $s = (x_1, x_2, ..., x_T)$ has corresponding label sequence $y = (y_1, y_2, ..., y_T)$, we integrate the label information and get a refined event keyword set for each sentence, denoted as $\hat{y} = (scene, character, action)$. $\hat{y}$ indicates the events in the current sentence.

A story paragraph example is presented in Table 1. The sentence-level detection results are listed. Event detection results of a story vary in different sentences and they are quite unbalanced. Only the 1st, the 8th and the 14th sentence have tokens indicating the story scenes. We need to infer that the first scene "field" should cover the sentence spans from the 1st to the 7th. And the scene changes to "river" in the 8th sentence and remains until the 13th one. Then it turns to "garden" and keeps the same until the end of the story. Similarly, we have to decide which character and action should appear in a sentence time span according to the paragraph information, even if nothing is detected in a specific sentence.

We mainly consider scene and character detection. An action may last from when it last emerged until the next action, such as running or driving. While it could also be short and happens within a sentence time (e.g. He sits down.). The determination of action continuity requires significantly more human knowledge and is beyond this paper's scope.

Extracted scene of a sentence is expanded to its neighbours in both forward and backward directions. At the scene boundaries, we follow the newly detected one. In this way, the story is divided into several scenes. Then we deal with characters within scenes. Normally, a character emerges at the first detected sentence and remains on the screen until the current plot ends.

6 Story Visualization

In this part, we calculate positions on the screen for each element. We define the position as $[left, top]$ in percentage relative to the top-left corner of the screen. Elements' positions are determined according to three constraints: 1) Meta data of the painting elements for the characters; 2) character number and significance in current time span; 3) history positions of the elements. The painting meta data of all elements include the following information:

- $(height, width)$: size of an element

The additional meta data of a painting scene are:

- $horizon$: distance from the horizontal line in a scene to the scene bottom. We use it as a safe line to arrange the feet of our characters; otherwise, a bear might float above the grassland, for example.

- point A: left point on the screen where we can locate a character.

story sentence (actions denoted with underlines)	scene	character
1. The sun beat down on the earth.	/	sun
2. Uncle cow was working in the field.	field	uncle cow
3. Beads of sweat were pattering down.	/	/
4. Seeing this, little elephant Lala quickly went to his side.	/	little elephant Lala, his
5. He fanned up big ears, and sent cool wind for uncle cow.	/	He, uncle cow
6. Uncle cow said with a smile:"Its so cool, thank you."	/	uncle cow
7. Lala replied happily:"No worries. No worries."	/	Lala
8. Grandma bear was washing clothes by the river,	river	Grandma bear
9. She was wiping sweat from time to time.	/	She
10. Lala saw it and fanned his big ears.	/	Lala
11. Grandma bear was not hot.	/	Grandma bear
12. She smiled kindly and said, "Good boy, thank you."	/	She
13. Lala replied:"No worries, no worries."	/	Lala
14. Little mouse was planting beans in the garden.	garden	Little mouse
15. Lala walked forward with enthusiasm and said,	/	Lala
16. "Little mouse, let me help you fan the wind."	/	Little mouse
17. "Thank you very much." said the mouse gratefully.	/	the mouse
18. Lala fanned her big ears again.	/	Lala
19. Suddenly he fanned the little mouse against the plantain leaf.	/	he, little mouse
20. Lala scratched her head shyly and said, "I'm really sorry."	/	Lala
21. Little mouse snort a laugh, and he said, "It's fun. It's fun."	/	Little mouse

Table 1: Example of extracted results for story "Big ears of the little elephant". (We have translated the Chinese story into English.)

- point B: right point on the screen where we can locate a character.

We calculate the character number to show on the screen in a time span and evenly distribute their positions based on the painting elements size and the horizon of the scene. Characters with high significance (talking ones or newly emerged ones) are placed near point A or B. If the character appeared in previous time spans, its position keeps the same or changes by minimal distance. The position should follow the equations:

$$top \leq 1 - height - horizon \quad (3)$$

$$left \leq 1 - width \quad (4)$$

$$\min ||top - top^{'}|| \quad (5)$$

$$\min ||left - left^{'}|| \quad (6)$$

where $top^{'}$ and $left^{'}$ stand for previous position of an element. If the element appears for the first time, Equation 6 and 7 are ignored.

As to the orientation setting, we initialize each character with an orientation facing towards the middle of the screen. Those who are talking or interacting with each other are set face to face.

Finally, we add a timeline to the story. Each event in the text is assigned a start time and an end time, so that it appears in the screen accordingly. Along with an audio track, the static images are combined to generate a story animation. The characters are mapped to corresponding elements with the detected actions if they are available (e.g., we have the elements when a character is saying). Dialogue boxes are added to show which character is saying. The painting elements are prepared in clip art style to make it more flexible to change them, as shown in Figure 2.

7 Experiment and Discussion

7.1 Experiment Setup

Data Collection: We collect 3,680 Chinese stories for children from the Internet[1]. The stories include 47 sentences on average. We randomly sample $10,000$ sentences from the stories and split them into three parts: training set (80%), testing set

[1]http://www.tom61.com
http://www.cnfla.com
http://story.beva.com
http://www.61ertong.com
(Our data are public copyrighted.)

Dataset	Train	Test	Dev
#sentences	8,000	1,000	1,000
#scene	5,896	711	664
#character	10,073	1,376	1,497
#action	15,231	1,949	2,023

Table 2: Dataset statistics.

Event	scene	character	action
Example	sea, forest...	fox, bear...	cry, run...

Table 3: Story events examples.

(10%), and development set (10%). We hired four experienced annotators to provide story events annotations. For each sentence, the annotators select event keywords and give them a category label of scene, character, or action. The words rather than event keywords are regarded as "others". We present the statistics of the collected corpus in Table 2.

Each sentence in the training and development set was annotated by one annotator for the sake of saving cost. But each sentence in the testing sets was annotated by three annotators independently. We calculate Fleiss' Kappa (Viera et al., 2005) to evaluate the agreement among annotators. For each token in a sentence, it is annotated as $y(y \in scene, character, action, others)$ by 3 annotators. The Fleiss' Kappa value is 0.580, which shows that the annotations have moderate agreement.

For story visualization, we hire two designers to design elements for storytelling. The elements include story scenes and characters (with different actions). Each frame of an animation consists of several elements. This mechanism is flexible for element switch and story plot development. We prepared 30 scenes and 600 characters, which have high frequencies in the collected stories. Some example animation elements are shown in Table 3.

Training Details: In the neural based methods, the word embedding size is 100. The LSTM model contains 100 hidden units and trains with a learning rate of 0.001 and Adam (Kingma and Ba, 2014) optimizer. The batch size is set to 20 and 50% dropout is used to avoid overfitting. We train the model for 100 epochs although it converges quickly.

Event	Method	Precision	Recall	F1
scene	Parser	0.585	0.728	0.649
scene	IDCNN	0.968	0.968	0.968
scene	BiLSTM	**0.973**	**0.974**	**0.973**
scene	BERT	0.931	0.918	0.924
chara	Parser	0.514	0.475	0.494
chara	IDCNN	0.829	0.758	0.792
chara	BiLSTM	0.831	0.758	0.793
chara	BERT	**0.833**	**0.853**	**0.843**
action	Parser	0.373	0.377	0.375
action	IDCNN	0.423	0.375	0.395
action	BiLSTM	0.442	0.400	0.420
action	BERT	**0.500**	**0.499**	**0.499**

Table 4: Sentence-level results comparison. (chara is short for character. BiLSTM and BERT represent BiLSTM-CRF and BERT-BiLSTM-CRF respectively.) We report the mean scores and conduct Tukey's HSD test. For scene extraction, the F1 score differences of all method pairs are statistically significant. So are that on character extraction (except the difference between BiLSTM and IDCNN). For action extraction, only the difference between BERT and Parser is significant.

7.2 Sentence-Level Evaluation

We compare the neural based models with a baseline based on parser. We first conduct word segmentation with Jieba (Sun, 2012) and part of speech (POS) annotation using Stanford CoreNLP Toolkit (Manning et al., 2014). Then we use dependency parser to extract events. For scene extraction, we find that most scenes in the childrens' stories are common places with few specific names or actions. Thus, we construct a common place dictionary with 778 scene tokens. We keep NP, NR, NT and NN (Klein and Manning, 2003) of POS tagging results and filter the scene tokens according to the scene dictionary. Dependency parser is employed to extract characters and actions. The subjects and objects in a sentence are denoted as the current story characters. The predicates (usually in terms of verbs or verb phrases) in the dependency tree are considered to contain actions of the corresponding characters.

The mean evaluation results over the test sets are shown in Table 4. The result shows that the BiLSTM-CRF method can achieve as high as 0.973 $F1$ score in scene extraction. The BERT-BiLSTM-CRF method can achieve 0.843 $F1$ score in character extraction, which is high too. But action extraction is the most difficult. Even

Sentence (actions denoted with underlines)	Scene	character
1. The chicken and duck walked happily by the lake.	lake	chicken,duck
2. The chicken and duck walked happily by the lake.	lake	chicken,duck
1. The rabbit's father and mother are in a hurry at home.	home	rabbit's father,mother
2. The rabbit's father and mother are in a hurry at home.	home	rabbit,father,mother
1. He walked into the big forest with his mother's words.	forest	He
2. He walked into the big forest with his mother's words.	forest	He,his mother
1. He said that he once donated money to mountain children.	/	he,children
2. He said that he once donated money to mountain children.	mountain	he,children
1. The rabbit walked and suddenly heard a deafening help.	/	rabbit
2. The rabbit walked and suddenly heard a deafening help.	/	rabbit

Table 5: Case study of sentence-level event extraction results.(1:Annotation, 2:Detection)

Event	Method	Precision	Recall	F1
scene	sentence	0.694	0.581	0.632
scene	paragraph	**0.878***	**0.837**†	**0.857**†
chara	sentence	0.705	0.763	0.733
chara	paragraph	**0.846**†	**0.987***	**0.911**†

Table 6: Paragraph-level event integration results. (chara is short for character.) $\dagger$ and $\star$ denote our improvements are significant in t-test with $p \leq 0.01$ and $p \leq 0.10$ respectively.

the best method BERT-BiLSTM-CRF can achieve 0.499 $F1$ score only, which is too low to use.

We conduct Tukey HSD significant test over all method pairs too. The results indicate that the neural methods are significantly better than the baseline based on parser in scene and character extraction. BERT-BiLSTM-CRF also significantly beats the parser baseline in action extraction. Among three neural methods, BERT brings significant improvements over the BiLSTM-CRF method in scene and character extraction. Only in scene extraction, BiLSTM-CRF is the best and the differences are significant.

Table 5 illustrates sample event extraction results. We can find that most of the story events are correctly extracted while there still exist a lot of biases. For example, some detected events do not actually happen in real but merely appear in the imagination or dialogues. (e.g. In verb phrase "heard a deafening help", the action is "heard", not "deafening".) Some serves as an adjective that modifies an character. (e.g. In noun phrase "mountain children", "mountain" does not indicate the current scene, but the children's hometown.)

7.3 Paragraph-Level Evaluation

In this evaluation, we focus on event switch detection. Take paragraph-level scene detection as an example. The story in Table 1 includes three scenes: field, river and garden, starting from the 1st, the 8th and the 14th sentence respectively. Paragraph-level event extraction is required to find the correct switch time and the event content. We compare simple extension of sentence-level results and paragraph-level event integration results (denoted as base and ours in Table 6).

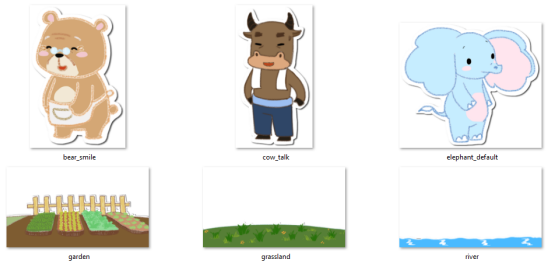
We randomly selected 20 stories from the collected corpus and manually annotated the scene and character spans. Scene keywords are mapped into 30 categories of painting scenes. Sentence-level scene results are extended in a way where the first sentence including the keyword is regarded as the start of the scene span and the previous sentence of next scene is denoted as the span end. For paragraph-level scene integration, scene spans are extended both in forward and backward orientation. Moreover, the dialogue contexts are ignored because the scene in a dialogue might not be the current one. It might be imagination or merely action of the past or the future. Other event information is also utilized as supplement, as the characters keywords might indicate specific scenes.

We calculate precision, recall and $F1$ value for event detection. A correct hit should detect both the event switch time and the right content. The results are listed in Table 6. As we can see, about 0.878% of scene switches are correctly detected. After story scene switch information extracted, it is used in paragraph-level character detection. Character switch is defined as the appearance and disappearance of a single character. The first time

(a) First scene: field

(b) Second scene: river

(c) Third scene: garden



(d) Painting elements

Figure 2: Visualized scenes and painting elements of story "Big ears of the little elephant"

when a character keyword is detected is denoted as the switch time of appearance. Scene switch is used as an indication of disappearance of the characters in that scene. Paragraph-level character detection reaches relatively higher accuracy than sentence-level character detection, with $F1$ score of over 0.91. T-test results indicate that our improvements are statistically significant.

7.4 Visualization Demonstration

Using the prepared 30 painting scenes and 600 characters, we are able to generate picture books for the collected 3680 stories, with 1.42 scenes and 2.71 characters in each story on average.

Figure 2 shows some story pictures and painting elements. More examples of video visualization results could be found on our website[2].

8 Conclusion and Future Work

In this paper, we propose a framework to address the problem of automatic story understanding and visualization. Story event extraction is extended from sentence level to paragraph level for continuous visualization. We collect children's story from the Internet and apply our framework to generate simple story picture books with audio.

[2]https://github.com/StoryVisualization/Demos

Currently, our story events include scenes, characters and actions. There is room for event extraction improvement. Furthermore, it is difficult to enumerate and compose an intimate action between characters, such as "hug", or a complex action, such as "kneeling on the ground". We plan to learn the various actions from examples, such as movies, in the future.

Acknowledgments

We would like to thank Qingcai Cui from Microsoft and Yahui Chen from Beijing Film Academy for providing helpful ideas, data and resources.

References

Eugene Charniak. 1972. *Toward a model of children's story comprehension.* Ph.D. thesis, Massachusetts Institute of Technology.

Hong-Woo Chun, Young-Sook Hwang, and Hae-Chang Rim. 2004. Unsupervised event extraction from biomedical literature using co-occurrence information and basic patterns. In *International Conference on Natural Language Processing*, pages 777–786. Springer.

K Bretonnel Cohen, Karin Verspoor, Helen L Johnson, Chris Roeder, Philip V Ogren, William A Baumgartner Jr, Elizabeth White, Hannah Tipney, and

Lawrence Hunter. 2009. High-precision biological event extraction with a concept recognizer. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 50–58. Association for Computational Linguistics.

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*.

Jon Doyle. 1979. A truth maintenance system. *Artificial intelligence*, 12(3):231–272.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Frederik Hogenboom, Flavius Frasincar, Uzay Kaymak, and Franciska De Jong. 2011. An overview of event extraction from text. In *Workshop on Detection, Representation, and Exploitation of Events in the Semantic Web (DeRiVE 2011) at Tenth International Semantic Web Conference (ISWC 2011)*, volume 779, pages 48–57. Citeseer.

Ting-Hao Kenneth Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. 2016. Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1233–1239.

Felix Jungermann and Katharina Morik. 2008. Enhanced services for targeted information retrieval by event extraction and data mining. In *International Conference on Application of Natural Language to Information Systems*, pages 335–336. Springer.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics.

Robert Kosara and Jock Mackinlay. 2013. Storytelling: The next step for visualization. *Computer*, 46(5):44–50.

Chang-Shing Lee, Yea-Juan Chen, and Zhi-Wei Jian. 2003. Ontology-based fuzzy event extraction agent for chinese e-news summarization. *Expert Systems with Applications*, 25(3):431–447.

Zhen Lei, Ying Zhang, Yu-chi Liu, et al. 2005. A system for detecting and tracking internet news event. In *Pacific-Rim Conference on Multimedia*, pages 754–764. Springer.

Fang Li, Huanye Sheng, and Dongmo Zhang. 2002. Event pattern discovery from the stock market bulletin. In *International Conference on Discovery Science*, pages 310–315. Springer.

Mingrong Liu, Yicen Liu, Liang Xiang, Xing Chen, and Qing Yang. 2008. Extracting key entities and significant events from online daily news. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 201–209. Springer.

Kwan-Liu Ma, Isaac Liao, Jennifer Frazier, Helwig Hauser, and Helen-Nicole Kostis. 2012. Scientific storytelling using visualization. *IEEE Computer Graphics and Applications*, 32(1):12–19.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

David McClosky, Mihai Surdeanu, and Christopher D Manning. 2011. Event extraction as dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1626–1635. Association for Computational Linguistics.

Yoko Nishihara, Keita Sato, and Wataru Sunayama. 2009. Event extraction and visualization for obtaining personal experiences from blogs. In *Symposium on Human Interface*, pages 315–324. Springer.

Masayuki Okamoto and Masaaki Kikuchi. 2009. Discovering volatile events in your neighborhood: Local-area topic extraction from blog entries. In *Asia Information Retrieval Symposium*, pages 181–192. Springer.

Jakub Piskorski, Hristo Tanev, and Pinar Oezden Wennerberg. 2007. Extracting violent events from online news for ontology population. In *International Conference on Business Information Systems*, pages 287–300. Springer.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the thirteenth conference on computational natural language learning*, pages 147–155. Association for Computational Linguistics.

Gerard Salton and Michael J McGill. 1986. Introduction to modern information retrieval.

Edward Segel and Jeffrey Heer. 2010. Narrative visualization: Telling stories with data. *IEEE transactions on visualization and computer graphics*, 16(6):1139–1148.

Hideo Shimazu, Yosuke Takashima, and Masahiro Tomono. 1988. Understanding of stories for animation. In *Proceedings of the 12th conference on Computational linguistics-Volume 2*, pages 620–625. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. *arXiv preprint arXiv:1702.02098*.

J Sun. 2012. jiebachinese word segmentation tool.

Hristo Tanev, Jakub Piskorski, and Martin Atkinson. 2008. Real-time news event extraction for global crisis monitoring. In *International Conference on Application of Natural Language to Information Systems*, pages 207–218. Springer.

Alex Tozzo, Dejan Jovanovic, and Mohamed Amer. 2018. Neural event extraction from movies description. In *Proceedings of the First Workshop on Storytelling*, pages 60–66.

Josep Valls-Vargas, Jichen Zhu, and Santiago Ontañón. 2017. Towards automatically extracting story graphs from natural language stories. In *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*.

Anthony J Viera, Joanne M Garrett, et al. 2005. Understanding interobserver agreement: the kappa statistic. *Fam med*, 37(5):360–363.

Akane Yakushiji, Yuka Tateisi, Yusuke Miyao, and Jun-ichi Tsujii. 2000. Event extraction from biomedical papers using a full parser. In *Biocomputing 2001*, pages 408–419. World Scientific.

Xin Zeng and Tan Mling. 2009. A review of scene visualization based on language descriptions. In *2009 Sixth International Conference on Computer Graphics, Imaging and Visualization*, pages 429–433. IEEE.

Deyu Zhou, Liangyu Chen, and Yulan He. 2014. A simple bayesian modelling approach to event extraction from twitter. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 700–705.

C Lawrence Zitnick, Devi Parikh, and Lucy Vanderwende. 2013. Learning the visual interpretation of sentences. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1681–1688.