Ablation Studies

In the following parts, we will carry out detailed ablation studies to better understand our model. In order to avoid training instability on small data and non-determinism of TensorFlow under GPU or TPU, except for reporting the best accuracy, we also report the average and standard deviation through 10 independent runs.[2] We ablate on Diff-Net without BERT to eliminate the effect by its powerful empirical performance to solely evaluate on our model. We will analyze the results based on the average score and its standard deviation, which is statistically stable and reliable. We first begin with the general components in our Diff-Net, and the results are shown in Table 3.

System	Test v1.0 Accuracy
Diff-Net	**77.8** (77.60 ± 0.12)
L1: SELU $\rightarrow$ Tanh	77.2 (77.04 ± 0.06)
L2: w/o cosine loss	77.2 (77.02 ± 0.20)
L3: w/o modified AoA	77.0 (76.93 ± 0.07)
L4: w/o match module	77.1 (76.85 ± 0.20)
L5: w/o discriminative module	76.9 (76.75 ± 0.21)
L6: AoA $\rightarrow$ dot product	76.7 (76.58 ± 0.11)
L7: w/o all binary features	76.0 (75.78 ± 0.12)

Table 3: Ablations on Diff-Net. The results are ordered by the descending average score. For reference simplicity, we label each experiment with L1 to L7.

When compared to the original AoA mechanism (L3), the modified AoA gives 0.67% improvements on average, indicating that the modified AoA is more powerful in the context of our model. Also, it demonstrates that the max-pooling is relatively superior at filtering noise and choose the most representative values in the vectors. If we replace the AoA mechanism to the simple dot product (L6), there is a significant drop by near 1%, suggesting that the AoA mechanism is helpful in precisely calculating attentions. By discarding the match module (L4), we see a significant drop in performance by 0.75%, while without the diff module (L5) gives an even lower score. This demonstrates that retrieving relevant information from the story as well as discriminating two endings are both important in this task, and combining these components yields further improvements. If we remove the cosine loss in training objective (L2), there is a slight drop in the performance as well as brings bigger fluctuation in results, indicating that adding additional loss could stabilize the experimental results, which demonstrates that separating latent semantic distance between two endings are helpful in this task. Also, as we can see that, without using any binary features (L7) will bring a significant drop in performance, which demonstrates that the matching features will help the model better recognize the alignment between the story and endings in the traditional models.

Recall that we have added three binary features to enhanced word embedding representation of the endings, especially when the training data is not enough. The ablation results are given in Table 4. Among three binary features, the most useful one is the end-story fuzzy matching feature, in the meantime, it could give slight improvements over the end-story non-fuzzy matching. The end-end matching feature brings moderate improvements which could be a remedy for providing mutual information of the endings.

[2]The average accuracy and standard deviation are shown in the brackets. Note that, due to the Non-Gaussian distribution of the results, `average + stdev` $\neq$ `max`.

System	Test v1.0 Accuracy
Diff-Net (all features)	77.8 (77.60 ± 0.12)
- w/o E-E Match	77.2 (77.05 ± 0.11)
- w/o E-S Match	77.0 (76.83 ± 0.12)
- w/o E-S Fuzzy Match	76.9 (76.78 ± 0.11)
- w/o all features	76.0 (75.78 ± 0.12)

Table 4: Ablations on using different binary features in word embedding (E: ending, S: story).

Settings	Diff-Net	BERT+Diff-Net	
	Test v1.0	Test v1.0	Test v1.5
Entire Story	**77.8**	**90.1**	**82.0**
- w/o 1st sent.	76.6 (-1.2)	90.0 (-0.1)	81.9 (-0.1)
- w/o 2nd sent.	77.2 (-0.6)	90.0 (-0.1)	82.0 (+0.0)
- w/o 3rd sent.	77.2 (-0.6)	89.6 (-0.5)	82.2 (+0.2)
- w/o 4th sent.	76.5 (-1.3)	83.6 (-6.5)	74.9 (-7.0)
Ending only	75.9 (-1.9)	80.6 (-9.5)	69.1 (-12.8)
Reverse story	77.5 (-0.3)	89.3 (-0.8)	81.4 (-0.5)
Random order	77.6 (-0.2)	89.3 (-0.8)	78.7 (-3.2)

Table 5: Quantitative analysis on using different proportion and sentence order of the story. The performance gap compared to the baseline is depicted in the bracket.

Discussion

While BERT, as well as our modifications, brings good performance on this task, there are still several questions that remain unclear.

- *Does models truly **understand or comprehend** the story?*
- *Is Story Cloze Test data (both v1.0 and v1.5) suitable for evaluating **story comprehension**?*
- *Except for the objective metric, in which aspects does BERT improve than the traditional neural networks?*

To investigate the questions above, we conduct comprehensive quantitative analyses to examine both models and datasets. We have an intuition that the last part of the story is critical for predicting the real ending. To verify this assumption, we discard each sentence in the story to see which sentence is of the most help in this task. Also, we set an experiment that does not use the story at all to see how much gain can we obtain by using the story information, to test the ability of *story comprehension*. The quantitative analysis results are shown in Table 5, and we get some unanimous and interesting observations.

Firstly, when we discard each sentence in the story, except for the last sentence in the story, the other sentences

seem to provide little help in this task regardless of models and datasets. In most situations, the first sentence is providing the background information and topic of the story, and we can see that it helps in finding the correct ending only in the traditional neural network model (w/o BERT). However, when it comes to BERT-based models, there is only little variance regardless of test v1.0 or v1.5. To our surprise, in test v1.5, though it was an evolved version of SCT, removing the second or third sentence in the story could weirdly *improve* overall performance, which was not expected. We suspect that the middle sentences are not the final state of the story, thus have little impact on the ending. Lastly, when removing the last sentence, the performance of all models in all sets decrease dramatically, which indicates that it is the most important in the story and provides key clues for predicting the real ending. That is, the *last-sentence bias* still exists in SCT v1.5. The following example shows the last-sentence bias in this task, where we could easily pick the real ending by only looking at the last sentence in the story.

[Story]
Janet worked hard to train for her wrestling meet.
When she got there her opponent seemed game.
They both tried their hardest.
It ended in a tie.
[Real Ending]
Janet was content with the result.
[Fake Ending]
Janet won the first place trophy.

Figure 3: An example of last-sentence bias issue. By only looking at the word 'tie' in the last story sentence, we can easily pick the real ending, as word 'won' in fake ending raises contradiction to the story.

Secondly, in Diff-Net, there is only a 1.9% decrease in the system performance without the presence of the story (ending only), indicating that the story does help in choosing the real ending, but the improvement is quite moderate. However, in BERT+Diff-Net, though the baseline increases a lot, as we can see that there is about 10% to 12% drop without the story in the test v1.0 and v1.5 data. This suggests that: 1) the traditional models focus less on the story and ending itself plays a key role. 2) The BERT-based model is better than traditional models in finding relations between the story and ending, as the input sequence is the concatenation by them and be fed into very deep transformer layers with self-attention mechanism.

Thirdly, to our surprise, reversing the story sentences or even randomly placing these sentences do not show a significant drop in the performance, which suggests that the order of the event sequence does not affect much in identifying the real ending in these datasets. Nonetheless, there is a significant drop in the test v1.5 compared to the counterparts, which demonstrate the test v1.5 does improve the evaluation on the narrative order of the story, but not that salient (only -3.2% in the accuracy). While **?** (**?**) discussed the importance of sentence ordering with respect to the story coherency, according to the results above, it seems not to be a crucial component in current *story comprehension* dataset.

Also, it can be inferred that current models are treating the sentences in the story as *discrete clues* rather than a temporal event sequence. Thus, we suspect the effect of using script knowledge for helping this task is quite limited, and we would investigate this in the future.

Conclusion

In this paper, we proposed a novel neural network model called Diff-Net to tackle the story ending prediction task. Our model could dynamically model the ending differences in three aspects and retrieve relevant information from the story. Also, we propose to use additional cosine objective function to separate the latent semantic distance between two ending representations. Experimental results on SCT v1.0 and v1.5 show that the proposed model could bring significant improvements over traditional neural baselines and BERT baselines. Except for the proposed model, we also carried out quantitative analyses on both traditional and BERT models and concluded that there is still a long way to go to achieve actual story comprehension.

As we indicated, the order of the story sentences does not affect the final performance much, in the future, we are going to verify our assumptions by introducing script knowledge to see if this could help in identifying real ending. Also, we would like to investigate the potential usage of unlabeled training data, such as training pre-trained models or constructing knowledge base for this task.

Acknowledgments

We would like to thank all anonymous reviewers and senior program members for their thorough reviewing and providing constructive comments to improve our paper. The first author was partially supported by the Google TensorFlow Research Cloud (TFRC) program for Cloud TPU access. This work was supported by the National Natural Science Foundation of China (NSFC) via grant 61976072, 61632011, and 61772153.

Deleniti a nostrum ab eligendi, asperiores aperiam amet molestiae deleniti voluptatem, explicabo mollitia quas, magni perferendis mollitia expedita voluptate quod.Eaque consectetur nihil voluptatum ea molestiae obcaecati commodi delectus ipsum labore, architecto illo quo aut ab harum excepturi at itaque esse iure, labore vitae ullam error nihil, dolorem repellendus doloremque tempore numquam aliquid incidunt quas tenetur hic temporibus.Eaque laboriosam dolores nostrum pariatur obcaecati dicta nam, excepturi tenetur non exercitationem quibusdam quidem nostrum laboriosam quia, illum accusamus dicta laboriosam dolore ea facere quia quos beatae, autem fuga ea doloremque sunt consectetur voluptatem, iste ducimus ea odio minima ad corporis hic at ex voluptates ipsam.Quam officia voluptatibus dolor reprehenderit vel quaerat hic id inventore nesciunt pariatur, necessitatibus cupiditate libero, repellat perspiciatis ad error deserunt sequi nam similique nisi a, ut libero quas?Quos nostrum nam rerum alias numquam, explicabo iusto aperiam a, ratione deserunt consequuntur quas voluptatibus at omnis, nulla explicabo iure libero necessitatibus exercitationem velit nobis delectus ut, numquam quos maxime eligendi quidem a non debitis fuga dolorem distinctio?Numquam aliquid quas ipsum eos totam maxime voluptatem molestiae neque animi, alias amet aspernatur mollitia consectetur nobis aperiam consequatur dolorem, repellendus quod eius