Unsupervised Hierarchical Story Infilling

Daphne Ippolito*
daphnei@seas.upenn.edu

David Grangier
grangier@google.com

Chris Callison-Burch
ccb@seas.upenn.edu

Douglas Eck
deck@google.com

Abstract

Story infilling involves predicting words to go into a missing span from a story. This challenging task has the potential to transform interactive tools for creative writing. However, state-of-the-art conditional language models have trouble balancing fluency and coherence with novelty and diversity. We address this limitation with a hierarchical model which first selects a set of rare words and then generates text conditioned on that set. By relegating the high entropy task of picking rare words to a word-sampling model, the second-stage model conditioned on those words can achieve high fluency and coherence by searching for likely sentences, without sacrificing diversity.

1 Introduction

Recent advances in language modeling have made considerable progress towards the automatic generation of fluent text (Jozefowicz et al., 2016; Baevski and Auli, 2019; Radford et al., 2019). This evolution has sparked the development of tools to assist human writers. For instance, Fan et al. (2018b) suggest generating short stories from high-level prompts, Clark et al. (2018b) study the interaction of human and language models for creative writing, and Peng et al. (2018) propose an interactive control of story lines. In addition, products such as Grammarly offer suggestions to improve grammar and wording (Hoover et al., 2015).

Our work is concerned with story infilling. We envision this task as a step towards a suggestion tool to help writers interactively replace text spans. Text infilling, a form of cloze task (Taylor, 1953), involves removing sequences of words from text and asking for a replacement. Compared to traditional left-to-right language modeling, automatic infilling interacts well with human text revision processes, which are rarely purely left-to-right. In the context of story generation, infilling should ensure (i) text fluency, (ii) coherence with the story line, and (iii) text which is not generic or obvious to a human. These three objectives require a delicate balance for modeling since fluency and coherence suggest preferring likely sequences, while novelty suggests preferring less likely sequences.

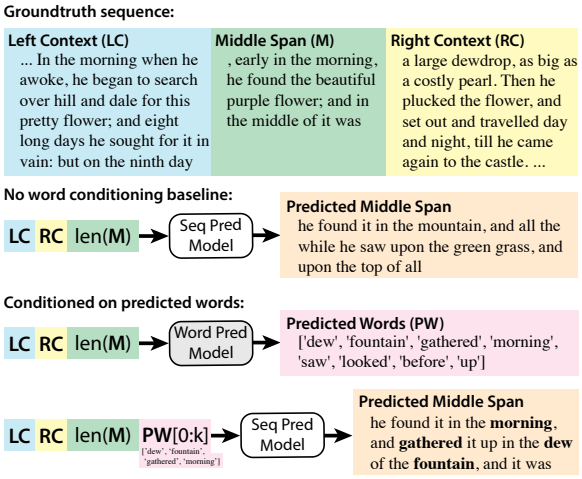

Figure 1: In the one stage baseline, the missing span is predicted given the context and the target length. In the two stage method, words that should go in the span are predicted in inverse frequency order. For visualization, the left and right contexts have been truncated.

We observe that recent conditional neural sequence to sequence models (Vaswani et al., 2017) have difficulty with this balance. As a solution, we propose to structure our cloze task in a hierarchical manner. In contrast to Fan et al. (2018b), we do not assume access to a supervised signal describing a hierarchy. We instead decompose our generation task by first randomly sampling from the high entropy part of the signal before generating the lower entropy part conditioned on the former. This decomposition is simple, yet powerful. The first model chooses rare words through ran-

*Work performed while a Google Student Researcher.

Proceedings of the First Workshop on Narrative Understanding, pages 37–43
Minneapolis, Minnesota, June 7, 2019.

dom sampling and the second model then uses a search algorithm to generate likely sequences conditioned on these words. Beam search in the second step allows better fluency (i) and coherence (ii), while conditioning with sampled words prevents novelty (iii) from being compromised.

We evaluate our proposal in the context of infilling passages from children's books and fairy tales. We compare vanilla transformer models with hierarchical alternatives, both through automated metrics and a human study. Our hierarchical method results in greater diversity in the generated text, without sacrificing quality. When we control for diversity, our method strongly outperforms the non-hierarchical baseline.

2 Related Work

Automatic Story Generation Computer-aided story generation has been a source of interest since the early days of NLP. Classical AI algorithms relied on symbolic and logical planning and graph construction (Klein et al., 1973; Meehan, 1977; Turner, 1993; Riedl and Young, 2006). Statistical methods have also been proposed (McIntyre and Lapata, 2009; Li and Riedl, 2015; Gatt and Krahmer, 2018). Recently, the field has been influenced by the success of (conditional) neural language models (Bengio et al., 2003; Schwenk and Gauvain, 2004; Bahdanau et al., 2015; Nallapati et al., 2016). Story generation with neural models include (Chourdakis and Reiss, 2017; Peng et al., 2018; Radford et al., 2019).

We build upon recent work that improves coherence in story generation by using hierarchical neural methods. These approaches allow reasoning at a higher level than words by considering a two-level hierarchy where a structuring representation conditions text generation. Martin et al. (2018) use sequences of events to structure generation while Jain et al. (2017) relies on sequences of short descriptions. Fan et al. (2018b) rely on writing prompts. Closer to our work, Clark et al. (2018a) condition on entity mentions. The training of these methods requires the availability of structuring labels which are either present in the training set (Fan et al., 2018b) or extracted by a separate system (Martin et al., 2018; Clark et al., 2018a). In our case, we avoid this step by considering rare words as the structuring signal.

Infilling Task Rather than generating an entire novel story, our goal is to replace text spans in an existing story to make progress towards interactive assistance for creative writers. Text infilling is known in linguistics as the cloze task (Taylor, 1953) and involves removing words or sequences of words from a text and asking a computer or a human to predict them. Existing work has used the masking of random words to build language models (Fedus et al., 2018) as well as contextualized word embeddings (Collobert et al., 2011; Devlin et al., 2018). Infilling of longer spans has been considered in work that explores bi-directional decoding for image captioning (Sun et al., 2017) .

3 Method

Our method predicts a variable length text span given a fixed length context from either side. We rely on the self-attentive Transformer model (Vaswani et al., 2017) with learned position embeddings, where the encoder takes the context as input and the decoder predicts the missing span. Architecture details and training parameters are in the Appendix. We use the subword tokenizer from (Vaswani et al., 2017), but report all statistics except perplexity in term of proper words. In addition to the context, we also condition our base model on the desired output length. We append to the input sequence a marker token denoting one of 5 possible length bins Fan et al. (2018a). Length conditioning lets us compare different models and decoding strategies with the same average generation length, thus avoiding length preference biases in human evaluation.

In our proposed approach, we decompose the generation task hierarchically, sampling a set of words desired for generation, before generating text that includes these words.

Word Prediction For each infilling instance, our model ingests the context data and predicts a sequence of subwords in frequency order, starting with rare subwords first. The word prediction model is a standard Transformer, for which we prepare the training data such that the target subwords are reordered by increasing frequency.

Our motivation for frequency ordering is twofold. Conceptually, rare words have a denser information content in an information-theoretic sense (Sparck Jones, 1972; Shannon, 1948), i.e., it is easier to predict the presence of common words given nearby rare words than the opposite. Practically, predicting rare words first allows us to interrupt decoding after a fixed number of steps, then

LC	were filled with anger, and decided not to go fishing again, but to wait for the next appearance of the fire. But after many days had passed without their seeing the fire, they went fishing again, and behold, there was the fire!	hand he held an iron club, which he dragged after him with its end on the ground; and, as it trailed along, it tore up a track as deep as the furrow a farmer ploughs with a team of oxen. The horse he	cave, whose mouth is beneath the sea. Here was a broad, dry space with a lofty, salt-icicled roof. The green, translucent sea, as it rolled back and forth at their feet, gave to their brown faces a
GT	And so they were continually tantalized. Only when they were out fishing would the fire appear, and when they	led was even larger in proportion than the giant himself, and quite as ugly. His great carcass was covered all over	ghastly white glare. The scavenger crabs scrambled away over the dank and dripping stones, and the loathsome biting eel, slowly reached
HIER-3	and there was a shout of joy from all the people who went fishing thither, but when they	rode was a lazy ox. He was a very ugly man. He was a man	faint intake of breath, whence it rose and curled, as it were, into the sea. And now it stretched
HIER-max	and thither they gathered together at a strong pace, for it was useless to go fishing at home, and when another shout	was missing stood in lazy work. You could see that he was a big, ugly ox,	shining intake of air, whence the black bear curled up on the surface of the water, and turned its head to look
BASE beam10	and they could not find it. They could not find it, and when the fire was	rode was a man of about thirty-five years of age. He was a tall man,	look of horror and horror. It seemed as if it would burst into a flood, and burst upon them, and burst
BASE sampling10	and the fire, which had been so long gone that many had not been in it for years, and when the fire	had driven was a little man of about the size of a man, with shaggy mane, and	deep, almost awful, impression, like that which was seen on a rock on a rocky beach. But the kangaroo did not stretch
BASE sampling	and at last there was a fierce fire! And at last Rosetta had an arrow, and when Oui	wheeled in without pausing to speak to me was a grotesque specimen of some repulsive animal. He was short of stature,	flood of radiance, sufficient to kill them utterly. [Illustration: It certainly had not a fairy named Serpent] The monster had cast
RC	returned they could not find it. This was the way of it. The curly-tailed alae knew that Maui and Hina had only these four sons, and if any of them stayed on shore to watch the fire while the others were out	with tangled scraggy hair, of a sooty black; you could count his ribs and all the points of his big bones through his hide; his legs were crooked and knotty; his neck was twisted; and as for his jaws, they were	out its well-toothed, wide-gaping jaw to tear the tender feet that roused it from its horrid lair, where the dread sea god dwelt. The poor hapless girl sank down upon this gloomy shore and cried, clinging to the kan

Table 1: Two qualitative examples with context extracted from fairytales. Left context (LC), right context (RC), ground truth center (GT), and the outputs from several methods are shown.

delegate the prediction of more common words to our second-stage model.

Word-Conditioned Generation The second-stage model, also a Transformer, is responsible for generating a text span given the surrounding context, a desired length marker, and a list of words predicted by the first-stage model. It takes as input the concatenation of these three signals.

At training time, we select a list of k words from the missing span to condition on, where k is sampled uniformly between 0 and half the target length. At inference, this model takes conditioning words from the word generation model introduced above. Interestingly, such a word list could be edited interactively by writers, which we defer to future work.

Training with a variable number of conditioning words allows us to choose the number of provided words at inference time. We observe that this choice needs to balance sufficient information to influence coherence and novelty in generated spans, while preserving some headroom for the second stage model to suggest its own common words and produce fluent text. Some examples of the unusual wording choices made when the second stage model is conditioned on all predicted words (HIER-max) can be seen in Table 1.

4 Experiments & Results

Experimental Setup We train on the Toronto Book Corpus (TBC) concatenated with Project Gutenberg, for a total of over 1.2 billion words after filtering our exact duplicate books. We withheld 5% of all books for validation and test.

Training examples consist of a 5 to 50 token-long target sequence, with 50 tokens of context on each side. We experimented with longer context windows but did not observe strong improvement on automated metrics. We do not force any alignment along linguistic boundaries, so context windows and gaps may start or end in the middle of a

Model	Decoding	Diversity		ROUGE-1	PPL	% Votes	p-value
		dist-1	dist-2	F1		against HIER-3	
BASE	beam10	.057	.218	0.29	16.61	48.75	0.82
BASE	sampling10	.058	.304	0.26	16.61	56.67	0.30
BASE	sampling	.101	.477	0.23	16.61	27.78	0.000025
HIER-max	sampling+beam10	.107	.442	0.24	4.22	28.33	0.00079
HIER-3	sampling+beam10	.104	.347	0.27	6.62	–	–

Table 2: Automated and human evaluation for our method (Hier) against baseline (base). Human evaluation reports A/B testing against Hier-3, along with chi-square test p-values.

sentence or even word.

Evaluation Automatic evaluation is performed on 10,000 spans of length 15-30 from our validation set. We report the sub-token perplexity of the reference and evaluate generation diversity with **dist-k**, the total number of distinct k-grams, divided by the total number of tokens produced over all examples in the validation set.

Three children's books were chosen from the validation set for human evaluation (Scott, 1921; Barrow, 1863; Vandercook, 1912). We hoped that the more concise prose in children's literature would make it easier for evaluators to quickly spot mistakes. We selected paragraphs of length 50 to 130 subwords, and randomly replaced a span of 15 to 30 subwords from anywhere in the paragraph.

Human raters were shown two instances of each paragraph, identical except for the selected span, which may have come from one model or another. The modified span was highlighted in each paragraph, and evaluators were asked which highlighted excerpt seemed better (more on-topic, exciting, and/or coherent) given the context. Further details about the task are in the the Appendix.

Results As our motivation is to generate diverse text without compromising on coherence and fluency, we evaluate the baseline non-hierarchical approach at different level of diversity by considering different decoding strategies. Conditional language models generate text word-by-word, either through beam search, i.e. approximating the maximum-a-posteriori sequence (Sutskever et al., 2014), or through sampling. Beam search often leads to repetitive, "safe" outputs, while random sampling results in more diverse outputs that mat suffer from fluency and coherence issues. While some work has incorporated a temperature parameter during random sampling to control the tradeoff between diversity and quality, we instead consider restricting sampling to the top-10 next words (sampling10) (Fan et al., 2018a) as preliminary experiments indicated this method produces higher quality outputs for equivalent levels of diversity.

Table 2 shows that as expected, sampling results in the richest diversity, beam search the poorest, and sampling10 falls between the two. In human evaluation, sampling10 and beam outperform or perform equivalently to our Hier-3 method, but have lower diversity. Unrestricted sampling performs much worse.

In our hierarchical approach (HIER), we achieve both diverse and fluent generation by using random sampling for the word prediction model, where diversity is more critical than fluency, and beam search for the second-stage model.

Table 2 evaluates HIER in two settings, conditioning on all words from the word prediction model or conditioned only on the first three predicted words. Human raters strongly prefer the model conditioned on only three words. We also show that humans rate generation of HIER-3 comparably to BASE/sampling10 while our model achieves much higher diversity (dist-1 and dist-2). Our model therefore achieves its goal of diverse and fluent outputs for story infilling.

5 Conclusions and Future Work

We show that taking a hierarchical approach to story infilling is an effective strategy for balancing fluent and coherent generated text with the diversity and interestingness necessary to build a useful tool for writers. Ultimately, we envision a fully collaborative system, where writers can upload a story and then solicit ideas from the computer on ways to rewrite specific parts. Writers will be able to choose between guiding generation by manually specifying words or concepts to be used, or taking suggestions made by the system.

Future work could investigate insertion-based architectures better suited to the infilling task (Stern et al., 2019), and the use of n-gram phrases instead of independent subwords as conditioning.

References

Alexei Baevski and Michael Auli. 2019. Adaptive input representations for neural language modeling. In *International Conference on Learning Representation (ICLR)*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representation (ICLR)*.

Fanny Barrow. 1863. *More Mittens: The Doll'S Wedding and Other Stories*. D. Appleton and Company, New York.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research (JMLR)*, 3(Feb):1137–1155.

Emmanouil Theofanis Chourdakis and Joshua Reiss. 2017. Constructing narrative using a generative model and continuous action policies. In *Proceedings of the Workshop on Computational Creativity in Natural Language Generation (CC-NLG 2017)*, pages 38–43.

Elizabeth Clark, Yangfeng Ji, and Noah A Smith. 2018a. Neural text generation in stories using entity representations as context. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2250–2260.

Elizabeth Clark, Anne Spencer Ross, Chenhao Tan, Yangfeng Ji, and Noah A Smith. 2018b. Creative writing with a machine in the loop: Case studies on slogans and stories. In *23rd International Conference on Intelligent User Interfaces*, pages 329–340. ACM.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Angela Fan, David Grangier, and Michael Auli. 2018a. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018b. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

William Fedus, Ian Goodfellow, and Andrew Dai. 2018. Maskgan: Better text generation via filling in the ____.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Bradley Hoover, Maksym Lytvyn, and Oleksiy Shevchenko. 2015. Systems and methods for advanced grammar checking. US Patent 9,002,700.

Parag Jain, Priyanka Agrawal, Abhijit Mishra, Mohak Sukhwani, Anirban Laha, and Karthik Sankaranarayanan. 2017. Story generation from sequence of independent short descriptions. In *SIGKDD Workshop on Machine Learning for Creativity (ML4Creativity)*.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Sheldon Klein, John F Aeschlimann, David F Balsiger, Steven L Converse, Mark Foster, Robin Lao, John D Oakley, Joel Smith, et al. 1973. Automatic novel writing: A status report. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Boyang Li and Mark Riedl. 2015. Scheherazade: Crowd-powered interactive narrative generation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Lara J Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark O Riedl. 2018. Event representations for automated story generation with deep neural nets. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Neil McIntyre and Mirella Lapata. 2009. Learning to tell tales: A data-driven approach to story generation. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 217–225. Association for Computational Linguistics.

James R Meehan. 1977. Tale-spin, an interactive program that writes stories. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 77, pages 91–98.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Conference on Computational Natural Language Learning (CoNLL)*.

Nanyun Peng, Marjan Ghazvininejad, Jonathan May, and Kevin Knight. 2018. Towards controllable story generation. In *Proceedings of the First Workshop on Storytelling*, pages 43–49.

Alec Radford, Jeffrey Wu Wu, Rewon Child, David Luan Luan, Dario Amodei, and Ilya Sutskever. 2019. Recognizing textual entailment: Rational, evaluation and approaches.

Mark O Riedl and Robert Michael Young. 2006. From linear story generation to branching story graphs. *IEEE Computer Graphics and Applications*, 26(3):23–31.

Holger Schwenk and Jean-Luc Gauvain. 2004. Neural network language models for conversational speech recognition. In *Eighth International Conference on Spoken Language Processing*.

Martin J. Scott. 1921. *A Boy Knight*. P. J. Kenedy & Sons, New York.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. *arXiv preprint arXiv:1902.03249*.

Qing Sun, Stefan Lee, and Dhruv Batra. 2017. Bidirectional beam search: Forward-backward inference in neural sequence models for fill-in-the-blank image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6961–6969.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Wilson L Taylor. 1953. cloze procedure: A new tool for measuring readability. *Journalism Bulletin*, 30(4):415–433.

Scott R. Turner. 1993. *Minstrel: A Computer Model of Creativity and Storytelling*. Ph.D. thesis, Los Angeles, CA, USA. UMI Order no. GAX93-19933.

Margarat Vandercook. 1912. *The Ranch Girls' Pot of Gold*. John C. Winston Company, Philadelphia.

Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416.

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 *Advances in Neural Information Processing Systems*, pages 5998–6008.

6 Appendix

7 Amazon Mechanical Turk Task

Our evaluation set consisted of 280 paragraphs selected from the evaluation dataset. For each question, evaluators were shown the same paragraph twice, with a highlighted span possibly altered by a model (Figure 2).

In our initial experiments, these questions were split into 20 HITs of 11 questions each. Ten of these questions compared generated text from the two methods of interest, while one other question was a honeypot, where one of the method outputs was replaced by the ground truth. However, after running multiple trial HITs, we found that the task was too hard for the average Turker, and performance on the honeypot question was close to random guessing.

We instead recruited two expert annotators familiar with reading antiquated English and with common language model mistakes to complete the HITs. In total we collected 60+ annotations per comparison task.

8 Model Parameters

All experiments were done with Transformer models implemented in the Tensor2Tensor framework (Vaswani et al., 2018). Important hyperparameters are shown below. All other hyperparameters were left at the Tensor2Tensor default.


```
{
    "attention_dropout": 0.1,
    "batch_size": 4096,
    "dropout": 0.2,
    "ffn_layer": "dense_relu_dense",
    "filter_size": 2048,
    "hidden_size": 512,
    "kernel_height": 3,
    "kernel_width": 1,
    "label_smoothing": 0.0,
    "learning_rate": 0.2,
    "learning_rate_constant": 2.0,
    "learning_rate_decay_rate": 1.0,
    "learning_rate_decay_scheme": "noam",
    "learning_rate_decay_steps": 5000,
    "learning_rate_warmup_steps": 8000,
    "num_heads": 8,
    "num_hidden_layers": 6,
    "optimizer": "Adam",
    "optimizer_adam_beta1": 0.9,
    "optimizer_adam_beta2": 0.997,
```

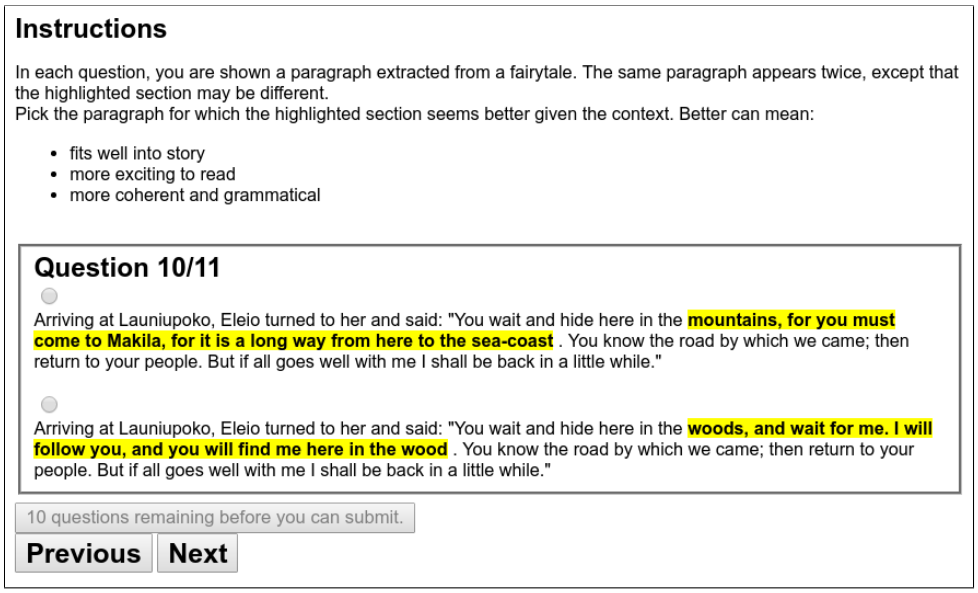

Figure 2: User interface for Amazon Mechanical Turk task.

```
    "optimizer_adam_epsilon": 1e-09,
    "pos": "emb",
    "self_attention_type": "dot_product",
    "train_steps": 1000000,
}
```