**Event-to-Sentence with BERT in Automated Story Generation**

Yingshan, CHANG

Instructor: Mark Riedl

Entertainment Intelligence Lab

College of Computing, Gatech

May-July, 2019

# Background

Automated story generation is the process of automatically choosing a sequence of events or actions and writing them into a sequence of sentences that can be told as a story. This process typically involves two stages: event-to-event and event-to-sentence. In event-to-event, we seek to recursively generated a sequence of events, an abstraction between words and sentences usually including a subject, an action, an object and some supplementary terms, which provide semantic information and logic of the story. In event-to-sentence, the events are expanded into a complete sentence, which should retain the semantic meaning of the event with some extra information to enhance readability.

This report mainly focuses on event-to-sentence with two proposed models based on pre-trained BERT: BERT for Filling the blanks and Editing-Writing Network. These two models utilize different capabilities of BERT, with the former taking the advantage of two pre-trained tasks, namely Masked Language Model (MLM) and Next Sentence Prediction (NSP), and the latter treats BERT as an encoder for the vector representation of a sentence, making use of BERT’s ability as a language model to generate word and sentence embeddings that incorporate the context, relationship between consecutive words as well as semantic information.

# BERT for filling the blank

Introduction

The basic logic is that, firstly, we create masked sentences by inserting [MASK]s between some tokens in the event. Since BERT strictly put one word at each [MASK], we need to let it explore multiple sentence patterns given the same event by putting different number of masks at different places. For each sentence pattern, BERT will do the Mask Language Model task by replacing each [MASK] by the most appropriate words in order to generate the most probabilistically correct sentence. In short, for each event, there will be multiple masked sentences, which become multiple completed sentences after BERT finishing its task.

Secondly, the Next Sentence Prediction task aims at choosing the best completed sentence among several candidates for each event. To be more specific, we take the previous context as “the first sentence”, which is fixed for a given event, and iteratively take every candidate as “the second sentence”, selecting the candidate sentence that scores the highest.

Methodology

1. Sentence-level N-gram

In each iteration, we provide the concatenation of the previously N-1 completed sentences as history, together with the next masked sentence as input to BERT. History and next masked sentence are separated by [SEP], a special separator token in BERT. The best completed sentence selected by NSP task is fed back to the history, treated as part of the input in the next iteration.

1. Generate Masked Sentence from Event

A certain number of [MASK]s can be inserted in the blanks between every two consecutive tokens of an event, including the very front and the very end. A parameter “max\_mask” specifies the maximum number of masks allowed in each blank, which is usually set to 2 or 3 in our experiments. Given an event, a list of masked sentences includes all possible outcomes by inserting [MASK]s in a certain way.

For example, given the event “he says”, the masked sentences should be:

{ “he says”,

“[MASK] he says”,

“he [MASK] says”,

he says [MASK]”,

“[MASK] he [MASK] says”,

“he [MASK] says [MASK]”,

“[MASK] he says [MASK]”,

“[MASK] he [MASK] says [MASK]” }

with max\_mask = 1.

1. Fill in the blank

Input is in the form of “[CLS] History [SEP] Masked\_Sentence [SEP]”, where [CLS] and [SEP] are the special tokens in BERT indicating start-of-sentence and separation. Then BERT generate a list of candidates by iteratively running through every masked sentence, replacing each [MASK] with a natural language word.

1. Pre-filtering before Scoring

We found that, with some masked sentences, BERT tend to put periods or repeated words in the blanks, which possibly suggests that it is hard for BERT to create a valid sentence according to the particular locations of [MASK]s. Thus, we discard the candidates with periods in the middle or consecutively repeated words before the scoring step, which also eliminate much unnecessary effort and computation.

1. Score Sentence

BERT is capable of capturing not only the relationship between words, but also that between sentences. The Next Sentence Prediction task demonstrates BERT’s sentence-level understanding of natural language. Given two sentences as input, it gives a score that measures how likely the second sentence can be the next sentence of the first one in a corpus. Thus, we use this function to decide which candidate is most likely to be the next sentence after a history of N-1 completed sentences. Besides, we also explored another two ways of scoring the sentence, which only make use of the general ability of BERT as a language model. The first is to concatenate the N sentences (N-1 history and 1 candidate) together and treat it as a “super sentence”. Then the score measures how likely this “super sentence” appear in natural language corpus. The second is to only score the candidate, ignoring the history. It turned out that both of them were generally worse than NSP. So all the scoring steps mentioned in the following article are based on NSP method.

Results

1. Consistency

Consistency illustrates how related the generated sentences are to the events. Since events are a higher-level abstraction of sentences, we want the generated sentences closely related the semantic meaning of the events, perhaps with some additional information that does not obscure, change or contradict to the primary message delivered by the events. It turned out that the generated sentences are highly-related to the event. This is because every word in the events must also appear in the generated sentences, serving as a restrictive force that makes BERT put only words consistent with their neighbors in the blanks.

1. Readability

Readability indicates the grammatic correctness and word coherence of a sentence. By just looking at the generated sentences and judging intuitively, the results are not too bad. In both MLM and NSP tasks, BERT was trained to make select words or sentences that could best make the results understandable and grammatically sound. However, there are occasional mistakes with part of speech, tense and word orders that could affect understanding.

1. Creativity

Creativity demonstrates how well BERT is able to add new information based on the event in order to make the sentences grammatically and semantically rich. But at first glance, BERT is too restricted by the words given in the event is not good at coming up with new information beyond what is conveyed by the event. In most case, BERT only adds adverbs that serve to indicate time and place, descriptive adjectives or conjunctions.

1. BLEU Score

BLEU, standing for Bilingual Evaluation Understudy, is an evaluating metric for machine translation systems. By measuring how many words or phrases in the candidate are actually present in the reference (ground truth), BLEU provides a method to compare the similarity between the candidate and the reference. In our task we used BLEU for a quantitative evaluation. We computed BLEU between BERT output and the ground truth for every example, then compute the average over the whole dataset, and the results are presented in the table below. Note that, since every word in the event will always appear in the generated sentence, BERT already gets some “hint” from the input. In order to eliminate the influence of those already-known words, we also calculated BLEU after removing the overlapping words included in the event from both candidates and references.

|  |  |  |  |
| --- | --- | --- | --- |
| Max\_mask |  | BLEU | BLEU  after removing overlap |
| 2 | Unit gram | 0.2816053101226246 | 0.09137763142993517 |
| Individual bi-gram | 0.0585804351289680 | 0.00746835963755909 |
| Cumulative BLEU-2 | 0.0896953873695871 | 0.01132960904430280 |
| Cumulative BLEU-3 | 0.0282076613883408 | 0.00188428725386206 |
| 3 | Unit gram | 0.2524006074972536 | 0.07249362320782772 |
| Individual bi-gram | 0.0498424847626522 | 0.007879239212844714 |
| Cumulative BLEU-2 | 0.0804431516128511 | 0.012133023098433388 |
| Cumulative BLEU-3 | 0.0302956797804649 | 6.3184578914129e-105 |

Unsolved Problems & Discussions

1. Time complexity

For a given event, the number of all possible masked sentence is determined by the number of tokens in the event as well as max\_mask. Suppose the number of tokens is k and max\_mask is m, then there are (m+1)^(k+1) masked sentences. The exponentially growing factor makes the computation time rocket up when the event become longer or max\_mask becomes larger. Perhaps it is not necessary to make a thorough exploration of all combinations of different locations or number of masks and a preprocessing step is needed to discard some bad masked sentences before the filling step.

1. Less of creativity

This problem can be attributed to the lack of information in the input. In the training process of MLM, only 20% of the words in a sentence are masks, while in our cases, more than 50% of the words are masks and there are usually only 3 to 5 real words in a masked sentence. The original MLM task emphasizes more on the ability of reading comprehension and filling in the blanks, while our new task emphasizes more on the ability of writing. Perhaps the compromised performance is due to the fact that what BERT had learned from the MLM task cannot be easily transferred to writing tasks. Therefore, further improvement calls for fine tuning of BERT.

# Editing-Writing Network

Introduction

1. Editing-Writing Network for Paper Abstract Writing

Our idea of Editing-Writing Network with BERT encoder is inspired by the paper: [*Paper Abstract Writing through Editing Mechanism*](https://www.aclweb.org/anthology/P18-2042)*.* It proposed an iterative text generation approach by which the model is able to generate a draft abstract first and come back to revise and improve the output, with the guidance from the input paper title as well as the previously generated abstract. According to the paper, this largely addressed the problem that typical RNN-based generation approach easily loses focus.

1. Original Model Architecture

https://camo.githubusercontent.com/7f3402d02031e5a84b6e0f729490d08471a378c9/68747470733a2f2f6561676c65772e6769746875622e696f2f696d616765732f77726974696e672d65646974696e672e706e673f7261773d74727565

* 1. Writing Network

The writing network is based on an attentive sequence-to-sequence network, with a bidirectional GRU as encoder and a GRU as decoder. In order to capture the relationship between the title and the generated draft, a soft-alignment attention mechanism is applied to provide a guide to the most relevant words in the title.

* 1. Editing Network

In the editing network, there are two separate encoders for the title and previous draft. The decoder, which shares weights with that in the writing network, also receives guidance from the previously generated draft in addition to the title, so that it can focus on more topically relevant concepts and further refine the output. A separate soft attentive function is used to compute the attentive context vector for previous draft, which does not share weights with title attention. Both title attention and previous draft attention are incorporated into the model by an Attentive Revision Gate that adaptively capture the relatedness of words in the title and words in the previous draft with every decoder step.

* 1. Multiple editing process can be involved and the output of the last editing process is considered as the final output.

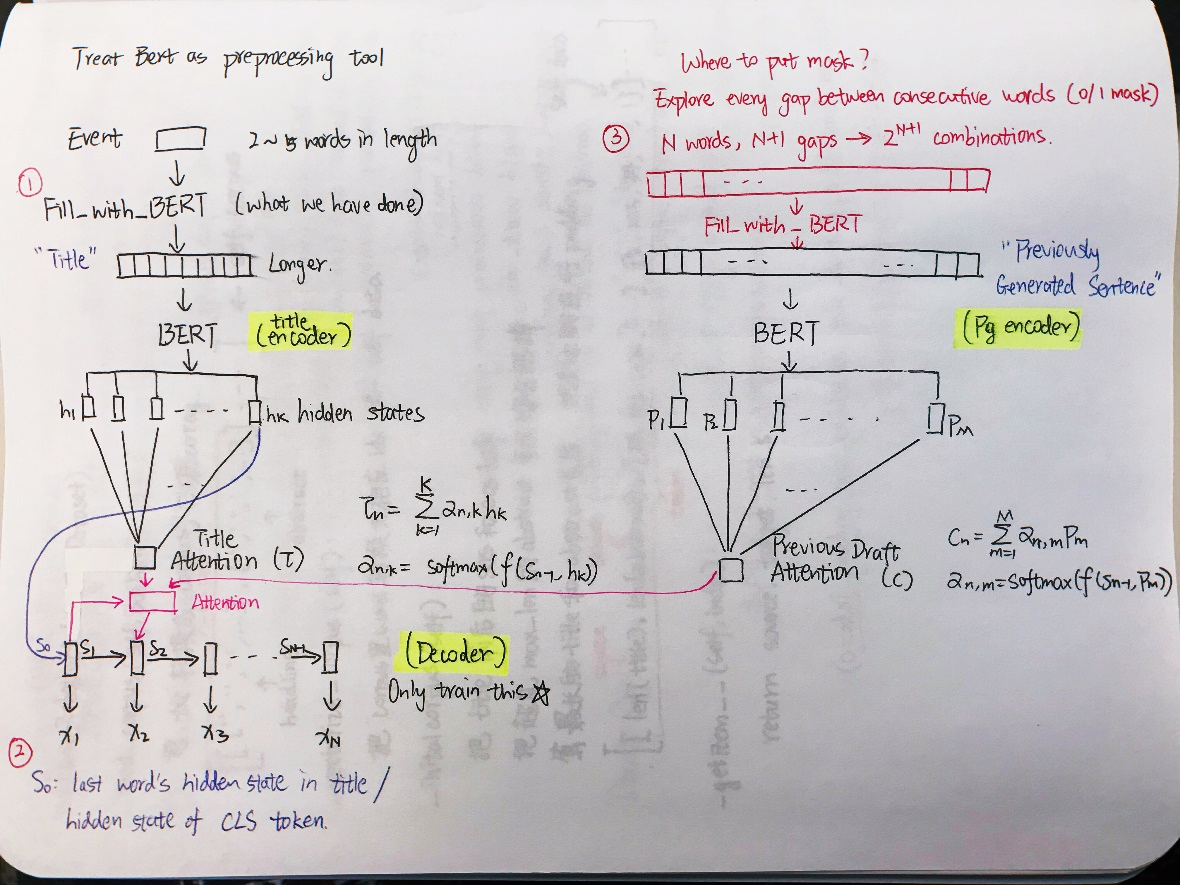
1. Major Change: LSTM encoder -> BERT encoder

BERT is a pre-trained model aiming at generating deep bidirectional representations from transformers by integrating the semantic meaning of both left and right context in all layers. The hidden vectors in the last several transformer layers can be viewed as encoding/embedding which should “encode” each word in the input sentence by taking both the semantic meaning of the words itself and its relatedness to the original sentence into consideration. Unlike the traditional RNN-based sequence to sequence encoder, BERT’s transformer is based on multiple layers of attention, which is superb in capturing the words’ meaning in a particular context. Thus, we thought replacing the sequence-to-sequence encoder described in the original paper by BERT is an idea worth trying.

Methodology

1. General Idea

Event-to-sentence is somewhat very similar to abstract writing from titles, since both aim at expanding a topic to a longer sentence or several sentences, keeping the output closely related to the given topic. Thus, the method proposed for paper abstract writing might be transferred to event-to-sentence by treating the event as “title” and the sentence as “abstract. The modified model architecture is demonstrated in the following graph.



1. Title Encoder

The input is the event with [CLS] and [SEP] in the front and the end according to BERT input requirement before being passed through the pre-trained BERT model. The hidden vectors in the last transformer layer is regarded as the encoding/embedding for each word in the input, since the last layer hidden states should be able to capture both short-term and long-term relatedness. The hidden vectors for [CLS] and [SEP] are discarded since they do not correspond to any particular word.

1. Previous Draft Encoder

This is largely similar to the title encoder, except that in the first iteration where the writing network writes the first draft, we use the event (“title”) as “draft number 0”. The reason is that, when we tried to use NONE as the “previous draft” for the first writing process, the output easily deviated from the event. Thus, by making “draft number 0” the same as the title, we force the model to write on the basis of the title, which might improve the consistency between the event and output.

1. Previous Title Encoder

This is an extra part compared to the network in the original paper, in order to promote a level of coherence between consecutive events. The encoding procedure is similar to the other two encoders.

1. Attentive Revision Gate

For every decoder step, we compute the soft attention context vectors for the encoder hidden states of current event, previous event as well as the previous draft, which serve as a summary of encoder outputs with emphasis on different parts. The three context attentive vectors are combined in the Attentive Revision Gate allowing the model to capture the correlation among the current event, previous event and draft, where the calculations are as follows:

**rm = σ(Wr,ccm + Wr,ττm + br)**

**zm = σ(Wz,ccm + Wz,ττm + bz)**

**pm = tanh(Wp,ccm + zm⊙(Wp,ττm + bp))**

**am = rm⊙cm + (1-rm)⊙pm**

**gm = σ(Wg,aam + Wg,uum + bg)**

**Am = gm⊙am + (1-gm)⊙um**

In the equations, **cm**, **τm** and **um** are the attentive context vectors of draft, current title, and previous title, respectively. W and b are learnable parameters. **Am** is the final output of the attentive revision gate, which is incorporated in the input at every decoder step.

1. RNN Decoder

We employ a GRU to generate the draft word by word. At each decoder step, the soft attention enables the decoder to focus on the most relevant words from the inputs. Besides, while the original paper used the last hidden state in the title encoder as the initial hidden state in the decoder, in our model, an average of hidden states of all words in the current event is considered as the decoder’s initial hidden state, for the average vector might serve as a “sentence embedding” that combines the general message conveyed by the current event.

Results

1. Consistency

In contrary to the BERT for filling-in-the-blank model, which ensures that all words in the event are going to appear in the generated sentence, the Editing-Writing Network is rewriting everything in the decoder. We found that the network tends to ignore the events when generating outputs. Although we initialized the “draft number 0” to be the same as the current event, in hope to give the network a foundation consistent with the event, it still turned out that the final output easily strays away from the event. The causes are not clear, perhaps some regularization mechanism, which reward the model when it generates the same/related words with the event, should be helpful.

1. Readability

By only looking at each sentence itself, the content can be understood in most cases, especially when the generated sentence is short. However, when it comes to relatively long output sentences, grammatic mistakes, word disorders or repeated words occur more frequently.

1. Creativity

At first glance, the lack of consistence also suggests that the model is somewhat “creative”, since it can generate outputs which are not restricted by the event and is able to come up with new contents, but a new problem is involved. After analyzing the outputs in depth, we found that many phrases or even a larger segment of the sentence in the outputs directly come from the training data, which means, the network may not learn how to capture the correlation between events and expected sentences, instead managing to memorize the training data, randomly extracting some pieces it has seen before, putting them together during testing.

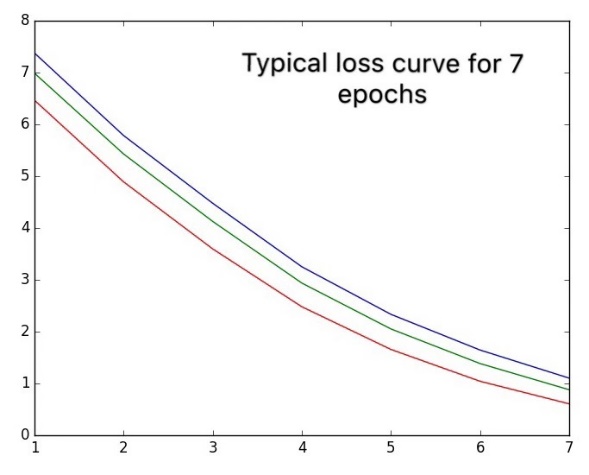
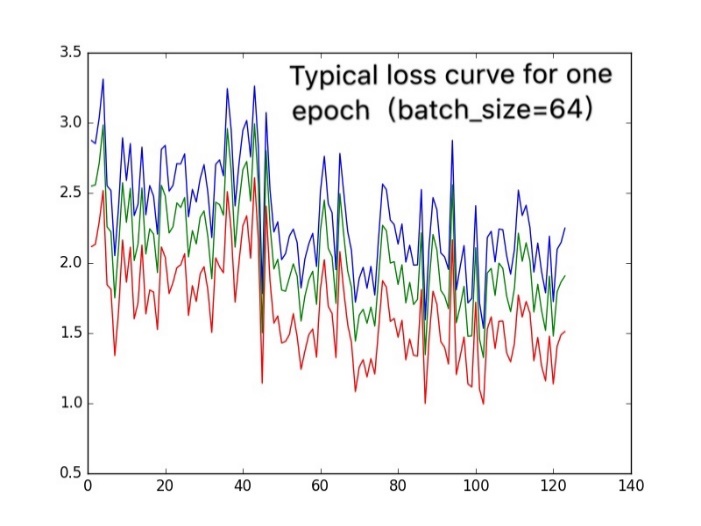
1. BLEU/METEOR/ROUGE Scores

We keep track of these scores of both training set and validation set during training, and stop if the average of these scores on the validation set begin to show a downward trend, so as to prevent overfitting. It is surprising that the scores are unusually high, nonetheless, just by looking at the outputs, the generated sentences lack much coherence and consistency with the inputs. This might be explained by the model generating many common words and phrases such as “the”, “go to”, “and then”, “he is”, etc. Moreover, we did not eliminate the effect of “hint” come from the input event, which may also lead to the high scores. The causes are unclear without further analysis.

|  |  |  |
| --- | --- | --- |
|  | Training set | Validation set |
| BLEU-1 | 9.577582140919736 | 8.392497643669403 |
| BLEU-2 | 3.5310604466814035 | 2.5028274334096685 |
| BLEU-3 | 1.5227979983392255 | 0.6867521385313033 |
| BLEU-4 | 0.7709967127135094 | 4.197911128262849e-05 |
| METEOR | 5.057141423661935 | 4.666231356240614 |
| ROUGE-L | 14.8859103013 | 14.2000575171 |

Unsolved Problems & Discussions

1. Unstableness of Training



The training process is highly unstable since we have integrated so many stuffs together. For one thing, there are violent fluctuations in the batch losses. Although there is a generally downward trend in the epoch loss, in each epoch there are always a few batches receive very high losses. I increased the batch-size from32 to 128 in order to smooth it out, but did not solve this problem entirely. From my understanding, the fluctuation has something to do with the fact that the examples in each batch are ordered rather than randomly chosen from all different parts of the training set, increasing the variation between different batches. To be more specific, in other machine learning tasks, the training examples are independent, so they are shuffled and randomly assigned to each batch. But in our case, each example is related to its previous and next example so that they should be arranged in a particular order.

For another thing, the validation loss usually stops to decrease only after 3-4 epochs of training. It is interesting to note that when the training is restarted, the loss will drop again. Possible explanation is that, when the model is saved, only the parameters are saved but the hyperparameters, especially the learning rate, is not. We were using “Adam” as the optimizer, which means the learning rate can be adaptively changed. But once the model is restarted, it is reinitialized to its initial value. Perhaps when the learning rate falls to a small value, it is hard for the model to escape a local minimum, leading to the termination of learning. Resetting learning rate gives it new energy to find better “low valley” in the optimization landscape. For further improvements, careful monitoring of the training process and smarter tricks of hyperparameter tuning are needed

1. Memorization vs. Writing ability

We found that during testing, the model tends to write exactly the same sentence or phrases in the training set, paying no attention to the consistency between its output sentence and the current event. This indicates a behavior of memorizing the exact sentences in the training data rather than learning the correlation between input and expected output. Maybe the model found it a better way to mimic the ground truth by memorizing rare words to phrases than by inferring the underlying correlation.

I would also like to view it from another point that, writing is a kind of “creation” task, but in the training process, we treat it as a “translation” task. The major difference between these two is that, in translation task, there is only one ground truth, or multiple ground truths with limited variations, while in creation task, the correct answers are unlimited and can be greatly varied. However, since it is nearly impossible to measure the goodness of an output under the assumption that there are a great many correct answers. Thus, on the current stage, we have to simplify the problem of evaluation to the measurement of similarity between the output with exactly one given ground truth. As a result, just like a writer who is forced to use the given sentence pattern, syntax as well as choice of words to write, the model’s creativity might be confined by the exact ground truth all the time and it might find memorizing a much easier shortcut to minimize the loss.

# Reference

1. Paper Abstract Writing through Editing Mechanism

<https://www.aclweb.org/anthology/P18-2042>

1. Event Representation for Automated Story Generation with Deep Neural Nets

<https://arxiv.org/pdf/1706.01331.pdf>

1. The BLEU Score: Evaluating Machine Translation Systems

<https://learning.oreilly.com/library/view/natural-language-processing/9781788478311/ch10s07.html>

# Appendix

BERT for Fill-in-the-blank outputs

Full results please see in the folder BLEU\_score&outputs

|  |  |  |
| --- | --- | --- |
| Event | Output (max\_mask=2) | Output (max\_mask=3) |
| daniel inform she about lords | carol and daniel to inform she is ##ll about lords ' plans . | before daniel is inform she is talks about the system lords ' coming meeting . |
| sun become hole  engine burn | the sun become hole - black . the engine burn . | the sun has become hole - black . the engine burn . |
| carter reach mars  carter find  tim reach mars | carter cannot reach mars safely . carter and carter find . will tim reach mars they dies . | carter reach the mars on station . carter and carter find . tim reach mars . |
| they break free from force | they break free from force . | they break free from force . |
| they try transfer [MASK] power | later again they try to transfer the power to earth . | they try transfer of power again . |
| they beam [MASK] by thor | they are a beam attack by the thor reactor . | they are beam down by the thor reactor . |
| holy planet collapse sun  sun create hole | holy war planet collapse into the sun . the sun fails to create a hole . | the holy earth planet collapse into the sun . sun radiation fail to create a hole there . |
| he inform they | will he inform them ? they are dead . | before he is to inform they are lost . |
| it takes machine [MASK] year | it take the machine a year to long . | can it take machine power control a year later . |
| they estimate  solution devise | they estimate . a solution devi ##se . | they cannot estimate . a solution is devi ##se . |
| they talk block  machine reverse device | they make talk block . the machine reverse device is failed . | then they talk block of control . the machine reverse device is failed . |
| [MASK] escape | the people escape planet . | they escape together . |
| they try escape  they escape  machine fire projectile on they | they try escape again . they escape . the machine fails to fire a projectile on they die . | they try escape again . they escape . the machine power can to fire projectile ##s on they escape . |
| carter start destroy  carter destroy machine  tim start destroy  tim destroy machine | carter has to start to and destroy . carter destroy the machine . tim has to start to and destroy . then tim has to destroy the machine again . | before carter they start again to destroy the control system . carter destroy the machine . tim they start again to destroy the control system . before tim starts they destroy machine power control . |
| carter beam [MASK] by ship  ship enters hyper space | the carter beam attack by the ship crash . the ship enter hyper - space . | the carter beam down by the ship crash . the ship enter hyper - space . |
| representative arrive  carol arrive  anna arrive  yu arrive  olivia arrive | the representative carter arrive there . carter , carol and carter arrive there . anna carter arrive there . carter and yu ##lia carter arrive . olivia carter arrive there . | the representative of carter arrive together . carter and carol carter arrive . anna carter arrive . carter and yu ##lia carter arrive . and olivia carter arrive . |
| balance destroy | the balance ##s destroy the planet . | the balance ##s destroy the system . |
| they reveal | they fail to reveal themselves . | they fail to reveal themselves . |
| they divide territory among they | they fail to divide territory among those they kill . | they that , they must divide territory among those they are to themselves . |
| that attack world  he plan attack | the forces that they will attack the world are destroyed . carter , he and to plan attack the planet . | those that attack the world are die . carter , he begins to plan to attack it . |
| god find inferno | god find inferno destroyed . | god begins to find inferno ##s . |
| they use weapon | they fail to use weapon technology . | they begin to use weapon weapons . |
| they ask | they fail to ask god . | they begin to ask the to ##s . |
| lords offer engine [MASK] they | the lords offer engine to they . | lords to offer engine power are they . |
| earth get territory  they defeat he | earth fails to get territory . when they defeat him , he dies . | earth begins to get their territory back . if they must defeat god , he is dead . |
| she propose | she propose marriage . | she begins to propose a marriage . |
| lords decline offer | the lords decline offer . | the lords decline offer of power . |
| machine head [MASK] towards shore  rich heads | the machine ships head back towards shore . the rich and people head home . | machine ships head back towards shore . the rich lords head home . |
| machine make thousand  machine make construct | machine ships make a thousand deaths . machine ships make money and construct ##s . | machine ships make a thousand deaths . machine ships make money and construct more ships . |
| she betray he | she betray ##s he dies . | she betray ##s he dies . |

BERT Editing-Writing Network outputs

Full results please see in the folder Writing-editing-Network-with-BERT\Writing-editing-network\data

|  |  |  |
| --- | --- | --- |
| Event | Generated Sentence (the 3rd draft) | Ground Truth |
| they find they surprise see they see man they look | They see them a message to them | They find what they are looking for and are also surprised to see a man, Zaddik, protecting the frightened Wraith |
| wraith feed on human wraith cause fear amongst villager | Finally of Pete transforms war, Replicators they want to speed a Human-form Human-form and begging Fifth to fend them | This is the Wraith that is feeding on the humans in the village and causing much fear amongst the villagers |
| they assume | They ask them | Immediately, they assume |
| they prepare kill they kill she | They estimated, them to Hala | They are prepared to kill her |
| they listen to <UNK> | They estimated, | They listen to his tale |
| man insist | Finally the engines could Hala | The man insists |
| wraith name <UNK> wraith discover by <UNK> | Several of the Replicators fire a block Lords Replicators a Human-form Human-form President completely 14, against Hala | The Wraith is named Ellia and was discovered by Zaddik at the site of the crash that the villagers described earlier |
| they learn | They try | They learn |
| he suppose | He is angry | He was supposed to |
| he bring himself himself kill she | He can't love him | He could not bring himself to kill her |
| he raise she by himself | He then tells her | He raised her by himself with compassion and love |
| she <UNK> thought | Instead Pete transforms | She dreads the thought of feeding off humans |
| human feed he create serum serum <UNK> need | Finally of the engines could Hala | She could not be responsible for the humans that are being fed upon because he has created a serum that suppresses the Wraith's natural need to feed on humans |
| <UNK> tell they | Instead they ask them | Zaddik tells them |
| they hear human feed | They try use forever with Hala | They hear of another human being fed on while Ellia is in the cave with them |
| team convince | The Ancient Lords offer | Initially wary of his claims, the team is convinced |
| <UNK> do it | It can't be it because | Ellia could not have done it |
| this mean | It is angry | This means |
| team excite about discovery | Finally of the engines could fend because they find out | The team is excited about this discovery |
| teyla ask <UNK> <UNK> help they discover wraith | Amaterasu then then reach Hala but find a Human-form Replicators who wants to fend her since Thor they reach Hala asylum | Teyla asks Ellia to help them discover the other Wraith |
| wraith connect through network | The Ancient Lords sent Hala | All Wraith are connected through a telepathic network |
| teyla plan use teyla use this they discover whereabouts | Amaterasu gets transforms one who then reach Hala | Teyla plans to use this to help them discover the whereabouts of the other Wraith |
| wraith head | Finally of the engines because they loves it to Hala | The other Wraith is heading |
| they have success they give direction | They are not angry they find a territories among a spider | They have some success with this and are given the general direction of |
| beckett call examine beckett examine serum | Amaterasu gets apologize authorized Replicator to Othala with Hala | Meanwhile, Dr Carson Beckett is called in to examine the serum |
| they believe | They ask them | They believe |
| beckett talk <UNK> talk | In the Replicator Replicator Replicator Carter Lords wakes Hala | Beckett and Zaddik are talking |
| he work | He gets angry | Iratus bug retrovirus that he has been working on |
| beckett mention | Thor is | Beckett mentions |
| wraith start as human | Several the System Lords offer war, the Replicators fire a Human-form Control between Earth could let be a Human-form Replicators power enough one to fend the Replicators on Hala | Wraiths started out as humans |