Creativity Evaluation Method for Procedural Content Generated Game Items via Machine Learning

Zisen Zhou[1], Zhongxi Lu[1], Matthew Guzdial[2] and Fabricio Goes[1]
[1]University of Leicester, Leicester, England, United Kingdom
[2]University of Alberta, Edmonton, Alberta, Canada
zz254@leicester.ac.uk, zl258@leicester.ac.uk, guzdial@ualberta.ca, lfwg1@leicester.ac.uk

2022 9th International Conference on Dependable Systems and Their Applications (DSA) | 978-1-6654-8877-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/DSA56465.2022.00042

***Abstract*—Procedural Content Generation via Machine Learning (PCGML) refers to methods that apply machine learning algorithms to generate game content. In particular, the generation of game item descriptions requires techniques to evaluate the similarity between items, and consequently their creativity. This paper improves the BLEU2vec text similarity evaluation technique by integrating it with Byte Pair Encoding (BPE) to capture the relevance of compound words in generated game item descriptions. This novel technique, called Split BLEU2vec, splits compound words into sub-words enabling their similarity evaluation. Our results show that when compared to BLEU2vec baseline, Split BLEu2vec is able to account for semantic embedding of compound words in item descriptions of the game Legend of Zelda.**

***Keywords*—Artificial Creativity; Procedural Content Generation; Creativity Evaluation**

I. Introduction

Items are inseparable parts of games, especially Role-Playing Games (RPG). From basic healing items and equipment to items that have special effects or progress the story, items play a key role in players' interaction with the world and can also serve as narrative devices that help players to immerse themselves in the game's fictional world.

Procedural Content Generation via Machine Learning (PCGML) refers to methods that apply machine learning algorithms to generate game content. Generation of content varies from game levels [1] and game quests [2] to game art [3] and game music [4]. Those methods can also be applied to the generation of creative game items. For an item description to be creative, it needs to be novel and valuable. Novel means that an item description has not existed in a game's asset, whereas value means this item description makes sense and can be used as development assistance or game asset.

Due to the complex nature of game item attributes such as their effect on the player's stats, lore, level, means of acquisition, and rarity, most previous techniques are rule-based
[5] to ensure the value of the generated items. On the other hand, machine learning techniques usually create complex novel items which their value is difficult to assess. In recent years, machine-trained metrics such as BERT [6] and factual correctness [7] are being used to evaluate Natural Language Generation (NLG) systems [8, 9]. However, major drawbacks of such approaches are unexplainable due to the BlackBox model, and potential bias that might be embedded in the training source [9].

There are other non-trained automatic evaluation metrics such as BLEU that evaluate the similarity between machine translation text with human reference texts [10]. Those metrics can serve as an evaluation of machine translation texts' novelty to human reference texts. These metrics are explainable, but they lack the capability of evaluating the similarity of the compound or original words that play an important role in the descriptions of game item.

In this paper, we propose a modified version of BLEU2vec [11], which we call Split BLEU2vec, that can account for the meaning of the various compound words in game item descriptions. It combines BLEU2vec, which is an enhanced version of BLEU, and Byte Pair Encoding (BPE) to split compound words into n-grams. To test our evaluation metric, we conducted a case study generating Legend of Zelda item descriptions via GPT-2 and also with a paraphrase dataset then used our evaluation metric to measure the similarity between the generated item descriptions.

The rest of the paper is structured as follows. First, we present the motivating problem for this paper. Second, we cover related work in this domain. Third, we present a general overview of our proposed method. Forth, we explain our experimental setup and present the results. Finally, we conclude and discuss potential future work for this research.

II. Motivation

Legend of Zelda and other Role-Playing Games (RPGs) tend to create unique compound words, such as 'dragonbone' which incorporate the semantics of 'dragon' and 'bone'. However, machine-generated item descriptions could contain words that are similar in meaning but not exactly like the original text. In this case, for example, if 'dragonbone sword' is in the original text and 'bone blade' is in generated text, techniques such as BLEU2vec would lose the semantic of 'dragonbone' and only compare 'sword' with 'bone' and 'blade', which makes it less

2767-6684/22/$31.00 ©2022 IEEE
DOI 10.1109/DSA56465.2022.00042

accurate in evaluating similarity. To account for this, we combined BLEU2vec with Byte Pair Encoding (BPE) [12] to split those original compound words into subwords from these special vocabularies before embedding. This approach attempts to get semantic value out of the compound words. For example, if we split 'biggoron' into 'big' and 'goron', at least 'big' will provide some semantic value to the compound word. This way 'biggoron' and 'big' will be assessed as similar, thus less novel.

Item Name: Slice of Cake

Original: ***Slices*** *of Cake, also known as Surprise-Filled Bread, are items in The Minish Cap.* ***Slices*** *of Cake can be purchased at the Bakery for 60 Rupees.*

Generated: They're found all over Termina, some being as thick as the walls of the Stone Tower SSB by The ***Slicekeeper****.*

TABLE I.

Example of BLEU2VEC and Split BLEU2VEC on a sentence with compound words. Bold text indicates similar words.

BLEU2vec	Split BLEU2vec
0.2477	0.2629

One of the issues that are not addressed in the BLEU2vec is that the word2vec model cannot encode words that haven't appeared in its training data [14]. For any such word, the word2vec model will return a vector of 0's, which can be an issue in analyzing generated item descriptions.

As an example, in Table I, we have found a pair of original and generated sentences from our evaluation results to compare with. Those sentences are not very similar semantically, but the generated item description contains the compound word **Slicekeeper**. As shown in Table I, Split BLEU2vec has a slightly higher score than BLEU2vec since BLEU2vec is not able to get the semantic meaning of the word 'Slicekeeper'. The embedding for this word is a vector of 0's, that is, unknown to the model. On the other hand, Split BLEU2vec was able to split 'Slicekeeper' into 'Slice' and 'keeper', and 'Slice' was similar to the 'Slice' word in the reference sentence, leading to a higher similarity score.

III. Related Work

This section presents the related work to our proposed similarity evaluation technique for game item descriptions.

Procedural Storytelling aims to generate story content, from scratch or alter based on the player's action [13]. This field relies on text generation and evaluation techniques to generate stories. In [14], the authors used a self-defined framework for the generation of a murder mystery plot. The framework is laid out as a set of procedural steps with corresponding rules, so the resulting artifact is a complete murder plot that the player can explore. Although items are introduced in the system, they are deliberately placed during development to avoid unnecessary distraction to the player, as opposed to generation via machine learning.

To evaluate closeness (or similarity) between machine translation text with human reference texts, the BLEU metric was proposed [10]. The machine translation community has been using this metric frequently to compare different translation systems. In recent years, the BLEU metric has been adapted to text generation communities for benchmarking such as evaluating the diversity of the generated text, by comparing each sentence to the rest of the generated document [15]. On the other hand, attempts have been made by Tättar to address the problem of BLEU not being capable of accounting for word similarity [11], by introducing word2vec into the BLEU system [16]. The work done by Tättar was mainly focused on the integration of BLEU and word2vec, thus they used the mature corpus of WMT 2015. Our work builds on top of Tättar and implements further to enable the evaluation of compound words.

Self-BLEU was a modified use of BLEU proposed by [17]. It adapts BLEU to measure output diversity of Natural Language Generation systems by using one of the generated texts as the hypothesis and all others as references. The lower the BLEU score indicates higher diversity. Self-BLEU is used in some other work for the evaluation of output diversity [18], while our work tries to evaluate the similarity of the text.

Lastly, we used Byte Pair Encoding (BPE) which is a text compression scheme to split words into sub-words. It was later adapted to find sub-words by [12] because of its strength in pattern matching.

IV. Split BLEU2VEC

Our Split BLEU2vec is adapted from [11], which uses word2vec to account for similarity within words, use cosine similarity to measure the closeness between words, use cosine similarity value as count, and aligning words to closest match greedily.

We started by modifying the original BLEU. It is intended to compare one sentence with another, but despite it can use multiple references, all these references need to have the same semantic meanings. This nature prevents us to evaluate semantic meaning with multi-sentence candidates and references, as the multi-sentence text in our case has a different semantic meaning per text. The simple modification we made was adding a sentence split feature to BLEU, so it can get n-grams from each of the sentences in a multi-sentence input and combine them in the end. The major drawback with this approach is that if some sentences are shorter than 3-gram or 4-gram, they might extend beyond their length. To account for this, in sentences

that are less than the required n-gram, the maximum available gram is added for evaluation.

Secondly, we integrated BPE into BLEU2vec to split the compound words into sub-words. The Split BLUE2vec algorithm steps are described as follows:

- All grams in the candidate sentence that are also in reference are considered accurate and assigned a weight of 1, including compound words.
- For all the remaining grams, they are embedded with the word2vec model. If a word has a vector of 0's, BPE splits the word into most likely sub-words. These sub-words are embedded individually, then their vectors are added together to form a vector for the compound word.
- Finally, the grams are greedily paired based on the maximum cosine similarity on their vector. No overlapping is allowed to reduce inflation of score.

V. Experiment Setup

In this section, we present the experimental setup to generate game item descriptions using GPT-2 and also the parameters used to evaluate the similarity of those sentences with Split BLEU2vec.

A. Model

Our game item descriptions creation system relies on fine-tuning the GPT-2 model. We chose GPT-2 as our generation model mainly because it is well-known for its domain adaptability, which we think is essential on our topic of item generation, because we are trying to create items that are relevant to the subject, in this case, Legend of Zelda-like items. We used a small model of GPT-2, which is 124M, due to limited resources.

B. Data

To fine-tune GPT-2, we first need a well-structured dataset. Our data was retrieved from one of the common Zelda-wiki sites, the Zelda fandom[1]. Our model has only retrieved data from the items page, specifically from the item description and "location and use" subsection. We believe that these two pieces of information are what is most relevant to an item's functionality, having less important information such as trivia reduces the noise in the dataset, thus the GPT-2 model might give us a more condensed result that purely focuses on item usage.

Assuming each item description is as an individual document with the item name as the document name, a tf-idf algorithm [19] is then applied to determine the most relevant keyword to each document. Only the sentences that contain these keyword remains. This is to further reduce the possible irrelevant information in the corpus, so each document is more condensed in providing a description specific to the corresponding item.

To structure the data for GPT-2, each sentence is then labeled with its corresponding item name. This gives GPT-2 a pattern of always associating an item description with its item name, so the output of a given prompt can be easily identified. The dataset is then split into a 9:1 ratio, with the majority of the dataset being the training set, and the remaining being the test set. The final test set contains 241 entries. We then use this dataset to fine-tune GPT-2.

C. Training

The training was conducted using the gpt-2-simple package of Python. The dataset was pre-processed to an array of Byte Pair Encoding (BPE) tokens before training. This is to reduce text complexity and split the meaning of sub-words. We used the adam optimizer because of its efficiency. We also used a batch size of 1 and a learning rate of 0.0001. After initial experiments, we determined that fine-tuning with 800 epoch enables the best model that minimizes the loss.

D. Generation

The temperature of generation used was 1. A dataset was generated by using each withheld test name as a prompt. As a separate point of comparison, we also generated a dataset using a paraphrasing model[2], each sentence of the set was a paraphrase of the original sentence in the test set. This later dataset can be viewed as a low-novelty, but a high-value dataset to compare with the GPT-2 generation dataset. We name the GPT-2 generation data as generation set and paraphrase data as paraphrase set.

E. Pre-processing

Before evaluation, generated datasets went through a preprocess to prepare for later n-gram counting and to reduce irrelevant variables such as tense and plurals. Firstly, the sentences were split by period, comma, and return, to retrieve sub-sentences from the sentence. This is to prepare for the later n-gram processing, as in the n-gram processing phase, n-grams are retrieved from sub-sentences rather than the whole sentence. Then we split each sub sentence to have a list of words and lemmatize them to reduce the effect of tenses and plurals. The result sentence can then pass on to evaluation.

[1] https://zelda.fandom.com/wiki/Main Page, April, 2022

[2] Sai Vamsi Alisetti, https://github.com/Vamsi995/Paraphrase-Generator, April, 2022

VI. Experimental Results

This section presents the similarity evaluation of the generated sentences by the trained model and paraphrases dataset.

TABLE II.

Comparison of BLEU, BLEU2vec, and Split BLEU2vec on the generation set.

	BLEU	**BLEU2vec**	**Split BLEU2vec**
mean	0.005174	0.05884	0.05919
std	0.01956	0.08989	0.09055

Table II shows a comparison between three methods of the evaluation of the generation set against withheld test sets. The result of BLEU2vec and our Split BLEU2vec is greater than BLEU by a noticeable margin, this indicates that the generated texts show some semantic relation to the original texts. On the other hand, the standard deviation of BLEU2vec and split BLEU2vec is more than 4 times as high as base BLEU, this can be due to the randomness and inflation of the score introduced by using word2vec.

From Table II, we also notice that our score is very close to the original BLEU2vec score, this is to be expected because the generation is not very controllable. Our proposed technique can evaluate sentences better only if compound words exist in the candidate and a word matching exists in reference, and vice versa. When the above criteria are not met, Split BLEU2vec behaves as BLEU2vec. Throughout the evaluation result, we noticed that Split BLEU2vec is consistently better than the BLEU2vec by a small margin.

TABLE III.

Comparison of BLEU, BLEU2vec, and Split BLEU2vec on the paraphrase set.

	BLEU	**BLEU2vec**	**Split BLEU2vec**
mean	0.6294	0.8494	0.8508
std	0.1494	0.08858	0.08774

Table III shows another comparison of three evaluation metrics on the evaluation of the paraphrase dataset against the withheld test. Despite the mean score of BLEU2vec and Split BLEU2vec be still greater than the baseline BLEU, the rate of difference has decreased significantly, this is due to the paraphrase dataset is mostly a re-ordered version of the original data with very few modifications of words. It is important to note that we observed that for most of the compound words, the paraphrasing model did not change them. The difference between BLEU2vec and Split BLEU2vec is still very small, this is largely due to the fact both metrics incorporate semantic meaning, thus non-identical n-grams in a candidate can be mapped to their closest meaning n-gram in a reference. Throughout the evaluation result, we noticed that Split BLEU2vec is consistently better than the BLEU2vec by a small margin. The standard deviation of BLEU in Table III was about 10 times higher than in Table II. We believe this is due to the randomness of the sentence's ability to be re-ordered. In sentences that can be re-ordered, BLEU is unable to find a match for a subset of n-grams because of its order-sensitive nature. On the other hand, BLEU2vec and Split BLEU2vec can still pair the most similar n-gram after re-order, thus less affected by the ordering. Furthermore, since Split BLEu2vec can also incorporate the meaning of compound words, it can find better matching for the n-gram than BLEU2vec, thus smaller is the standard deviation. As mentioned in previous sections, our Split BLEU2vec metric serves as a novelty assessment for the generated item description to original item description.

In tables II and III, we observe that Split BLEU2vec has scores just a bit higher than BLEU2vec, this indicates that our method has been implemented properly, because compound words are not abundant, thus they should make less impact in the final score. Our Split BLEU2vec evaluates the similarity not just of ordinary words, but also compound words, thus it is more suitable for evaluating novelty in game item text descriptions.

VII. Conclusion

This paper presented a compound words aware similarity evaluation in game item descriptions called Split BLEU2vec. Our results show that when compared to BLEU2vec baseline, Split BLEU2vec is able to account for semantic embedding of compound words in item descriptions of the game Legend of Zelda. Our Split BLEU2vec metric is suitable for determining novelty for text generation models that have compound words in their references or generated texts.

For future work, one aspect is to improve sub-word embedding. Currently, sub-words are straightly embedded for this initial study, but some sub-words lose semantic meaning if they are not completed to a whole word or substituted with a word of close meaning. Another direction would be applying Split BLEU2vec to other domains such as medical corpus, as most Latin terms used are compound words.

References

[1] Z. Zhou and M. Guzdial, "Toward co-creative dungeon generation via transfer learning," in *ACM 16th International Conference on the Foundations of Digital Games (FDG) 2021*, 2021, pp. 1–9.

[2] E. S. de Lima, B. Feijo´, and A. L. Furtado, "Procedural generation of quests for games using genetic algorithms and automated planning." In *SBC SBGames*, 2019, pp. 144–153.

[3] M. Guzdial, D. Long, C. Cassion and A. Das, "Visual Procedural Content Generation with an Artificial Abstract Artist," *ACC Proceedings of ICCC computational creativity and games workshop*, ICCC' 17, Atlanta, Georgia, USA, 2017.

[4] D. Plans and D. Morelli, "Experience-driven procedural music generation for games," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 3, 2012, pp. 192–198.

[5] C. K. On, N. W. Foong, J. Teo, A. A. A. Ibrahim, and T. T. Guan, "Rule-based procedural generation of item in role-playing game," *Insight Int. J. Adv. Sci. Eng. Inf. Technol*, vol. 7, no. 5, 2017, p. 1735.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[7] C. Zhu, W. Hinthorn, R. Xu, Q. Zeng, M. Zeng, X. Huang, and M. Jiang, "Enhancing factual consistency of abstractive summarization," *arXiv preprint arXiv:2003.08612*, 2020.

[8] A. Celikyilmaz, E. Clark, and J. Gao, "Evaluation of text generation: A survey," *arXiv preprint arXiv:2006.14799*, 2020.

[9] P.Narang, V. S. Ajay, and M.Himanshu. "Hybrid Metaheuristic Approach for Detection of Fake News on Social Media," International Journal of Performability Engineering, vol. 18, no. 6, June 2022, pp. 434-443.

[10] C. Leiter, P. Lertvittayakumjorn, M. Fomicheva, W. Zhao, Y. Gao, and S. Eger, "Towards explainable evaluation metrics for natural language generation," *arXiv preprint arXiv:2203.11131*, 2022.

[11] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *ACL Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ser. ACL '02. USA: Association for Computational Linguistics, 2002, pp. 311–318.

[12] A. Tättar and M. Fishel, "bleu2vec: the painfully familiar metric on continuous vector space steroids," in *WMT Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 619–622.

[13] Y. Shibata, T. Kida, S. Fukamachi, M. Takeda, A. Shinohara, T. Shinohara, and S. Arikawa, "Byte pair encoding: A text compression scheme that accelerates pattern matching," Dept., Kyushu Univ., Kyushu, Japan, Tech. Rep. DOI-TR-CS-161, Apr. 1999.

[14] C. Su, and D. Huang. "Hybrid Recommender System based on Deep Learning Model." International Journal of Performability Engineering, vol. 16, no. 1, January 2020, pp. 118-129.

[15] M. Hendrikx, S. Meijer, J. Van Der Velden, and A. Iosup, "Procedural content generation for games: A survey," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 9, no. 1, Feb. 2013, pp.1-22.

[16] A. Stockdale, "Cluegen: An exploration of procedural storytelling in the format of murder mystery games," in *AAAI Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*, AIIDE-16, Burlingame, California, USA, 2016.

[17] Y. Zhu, S. Lu, L. Zheng, J. Guo, W. Zhang, J. Wang, and Y. Yu, "Texygen: A benchmarking platform for text generation models," in *The 41st International ACM SIGIR Conference on Research Development in Information Retrieval*, ser. SIGIR '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1097–1100.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[19] Y. Zhu, S. Lu, L. Zheng, J. Guo, W. Zhang, J. Wang, and Y. Yu, "Texygen: A benchmarking platform for text generation models," in *The 41st International ACM SIGIR Conference on Research Development in Information Retrieval*, ser. SIGIR '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1097–1100.

[20] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," *arXiv preprint arXiv:1904.09751*, 2019.

[21] J. Ramos., "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, ICML-2003, Washington D.C., USA, vol. 242, 2003, pp. 133-142.