Creativity Embedding: a vector to characterise and classify plausible triples in deep learning NLP models

Isabeau Oliveri
Politecnico di Torino
isabeau.oliveri@
polito.it

Luca Ardito
Politecnico di Torino
luca.ardito@
polito.it

Giuseppe Rizzo
LINKS Foundation
giuseppe.rizzo@
linksfoundation.com

Maurizio Morisio
Politecnico di Torino
maurizio.morisio@
polito.it

Abstract

English. In this paper we define the creativity embedding of a text based on four self-assessment creativity metrics, namely *diversity, novelty, serendipity and magnitude*, knowledge graphs, and neural networks. We use as basic unit the notion of triple (*head, relation, tail*). We investigate if additional information about creativity improves natural language processing tasks. In this work, we focus on triple plausibility task, exploiting BERT model and a WordNet11 dataset sample. Contrary to our hypothesis, we do not detect increase in the performance.

Keywords - Creativity Embedding; Creativity Metric; NLP; Creativity Evaluation; Triple; Knowledge Graph; BERT.

1 Introduction

Current conversational agents have emerged as powerful instruments for assisting humans. Oftentimes, their cores are represented by natural language processing (NLP) models and algorithms. However, these models are far from being exhaustive representation of reality and language dynamics, trained on biased data through deep learning algorithms, where the flow among various layers without could result in information loss (Wang et al., 2015). As a consequence, NLP techniques still find it challenging to manage conversation that they have never encountered before, reacting not efficiently to novel scenarios.

One way to mitigate these issues is the integration of structured information, which knowledge graphs are one of the best-known systems for representing them. The most prominent example is the Semantic Web (Berners-Lee et al., 2001), where the information is represented through linked statements, each one composed of *head,relation,tail*, forming a *triple* (Figure 1). This semantic embedding allows significant advantages such as reasoning over data and operating with heterogeneous data sources.

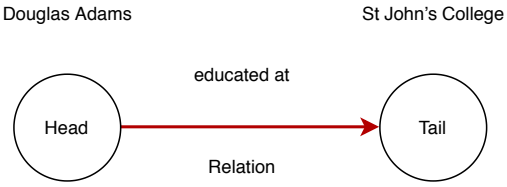

Figure 1: The triple (Douglas Adams, educated at, St John's College), from Wikidata knowledge base (Vrandečić and Krötzsch, 2014), is an example of statement.

Integration of structured information is not the only method that literature provides us to improve NLP techniques. Previous researches pointed out that analysis of creativity features could improve self-assessment evaluation, with benefits for solutions generated and inputs understanding (Lamb et al., 2018; Karampiperis et al., 2014; Surdeanu et al., 2008). We specify that in this work creativity is intended as capability to create, understand and evaluate novel contents. The concepts of Creativity AI have been discussed in their interconnections with the Semantic Web (Ławrynowicz, 2020), generalizable to knowledge graphs. Kuznetsova et al. (Kuznetsova et al., 2013) define quantitative measures of creativity in lexical compositions, exploring different theories, such as divergent thinking, compositional structure and creative semantic subspace. The crucial point is that no every novel combinations are perceived creative and useful, distinguishing creativity perceived in unconventional, uncommon or

”expressive in an interesting, imaginative, or inspirational way”.

Despite it is made clear the interest of the scientific community in exploring this direction, little research is conducted over creativity in the NLP field. The results and the considerations made by Kuznetsova and Ławrynowicz, led us to investigate the possible correlations between improvements in NLP tasks and creativity, with a particular focus on self-assessment. In this paper we introduce a novel approach for supporting deep learning algorithms with a mathematical representation of creativity feature of a text. We named it creativity embedding and based it on metrics of self-evaluation creativity over graph knowledge base.

2 Approach

2.1 Self-assessment creativity metrics

When humans face a problem they never encountered before, they usually perform a self-assessment procedure respect their previous knowledge and context, generally voting for the best solution. Following the example reported in Figure 2, we can imagine that a person has to describe the colour of a grey desk. He does not remind the name of the colour at that time, and performs a creative process. He use a metaphor to describe the grey colour of the desk, referring to the stereotype colour of a ”mouse”. This metaphor is widely accepted, and the colour would be ideally understand by the interlocutor. If in place of ”mouse” the random term ”mask” is used, the meaning will not probably received if not particular context or knowledge is shared between the person and the interlocutor, resulting in a not effective creative process. To emulate this self-assessment procedure, we propose metrics inspired by the related-concept literature, such as recommender systems (Monti et al., 2019) and machine learning (Pimentel et al., 2014; Ruan et al., 2020). The knowledge is represented by a graph of items interconnected by their relation (triples).

We define four metrics, namely diversity (1), novelty (2), serendipity (3), and magnitude (4). In these metrics we make use of a similarity function. In fact, to define the similarity (or the diversity, from another angle) between two or more items, we need a method and a representation that allows us to define a distance

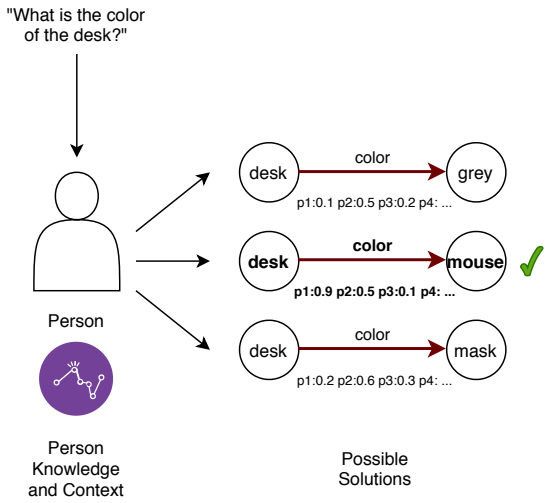

Figure 2: A person produces different solutions to answer a question. Therefore he performs a self-assessment procedure, taking into account several parameters p based on its knowledge and the context. Finally, he chooses the possible best solution. Parameters are expressed as numbers, for simplicity.

between them. In the literature, there is no fixed notion of similarity. However, a common strategy for texts is transforming words and sentences in vectors, taking in account and keeping their distributional properties and connections. Subsequently, mathematical distance functions are applied. The similarity function could defines a semantic similarly function between two items (words or sentences) under these conditions. For prompt understanding, we anticipate that in our experiment we use cosine similarity function and BERT vectors (embeddings) as words representation, as will be discussed in following sections. Nevertheless, thus defined metrics could be computed with different item vector representation and similarity function, as long as it is adopted a similarity function with output domain [0,1], with high value for high similarity.

Diversity (1) represents the semantic diversity between the head h_T and tail t_T of the triple T. This information tells how these two elements are not semantically close. It could be considered as T internal semantic diversity.

$$div(T) = 1 - similarity(h_T, t_T) \qquad (1)$$

Novelty (2) of a triple T is its average semantic diversity respect others triples in the context.

Context *C* is the sub-graph of triple obtained by traversing the paths of length *p* in the knowledge graph, starting from the triple h_T under examination, collecting *n* nearest triples. It could be considered as external semantic diversity of T respect to the context *C* retrieved.

$$nov(T) = \frac{1}{n}\sum_{i=1}^{n} 1 - similarity(T, C_i) \quad (2)$$

Serendipity (3) is here intended as the semantic novelty of the triple T, taking into account the *s* most novel triples considering the knowledge graph (refined context *S*). It could be considered as T novelty relevance.

$$ser(T) = \frac{1}{s}\sum_{i=1}^{s} 1 - similarity(T, S_i) \quad (3)$$

Magnitude (4) outlines the rarity of the triple, ranking *rk* each component of the triple by the number of its occurrences over the total number of items in the knowledge graph. The ranking function thus defined has an output domain [0,1].

$$mag(T) = \frac{rk(h_T) + rk(rel_T) + rk(t_T)}{3} \quad (4)$$

2.2 Creativity Embedding

There were no annotated datasets on the creativity characteristics of interest. For this reason, a direct comparison with the ground truth was hampered. To overcome this obstacle, we indirectly measured the effectiveness of this approach by applying it to an external model and judging the results on the triple plausibility task (Yao et al., 2019; Wang et al., 2018; Wang et al., 2015; Padó et al., 2009). The triple plausibility task consists of classifying a dataset's triples in plausible or not plausible classes, comparing the result respect to the ground truth. We choose this task to perform an indirect evaluation of our proposal, rely on the correlation between plausibility and creativity (Lamb et al., 2018), as plausibility could represent a positive outcome of an effective creative process. The current trend in machine learning and natural language processing models pushes the use of mathematical representation of meaningful information utilising vectors, commonly known in this field as embeddings. For these reasons, we outline and train a neural network using the computed ground truth to predict creativity values, and define as creativity embedding the weight of last hidden layer. This creativity embedding can be added and adapted in its dimension. Stated the above concepts, we define the subsequent research questions.

Research Question: *A creativity embedding extracted from the creativity neural network could improve triple plausibility classification in deep learning models?*

3 Model Architecture

3.1 BERT

We select Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) as a model for investigating the effects of creativity embedding, due to its flexibility and modularity, as well as being state of the art for various NLP tasks. The BERT model could be divided into three main parts: preprocessing of the input, stack of transformer layers, and other layers on top to perform a particular task - typically a classifier. A stack of *Transformers* forms the BERT core. A transformer exploits the attention mechanism to learn the contextual relationship between sentences and words input. The input is not considered in one direction, but figuratively in all ones at one time, defining the context of a word considering the entire surrounding words. The model is trained with a sort of play, where some words or entire sentences are masked, and the model has to predict them. We do not modify the core of the model; we are more interested in the preprocessing part, where we will inject the creativity embedding, as explained in the next section.

3.2 Creativity Neural Network and Creativity CLS Embedding

The outline of the architecture proposed for the task is shown in Figure 3. In the lower part, the triple flows through the BERT model. We used a modified tokenization technique of Knowledge Graph BERT (KG-BERT) (Yao et al., 2019), adapted for the structure of the triple. The triple is split in tokens respect the BERT vocabulary of known words. Special tokens are included in the sequence, classification (CLS) and separator (SEP) tokens. CLS corresponding embeddings are in charge of representing the sentence mathematically, and SEP tokens that separate different sentences. On the KG-BERT version for triple plausibility, SEP is used to separate head words from

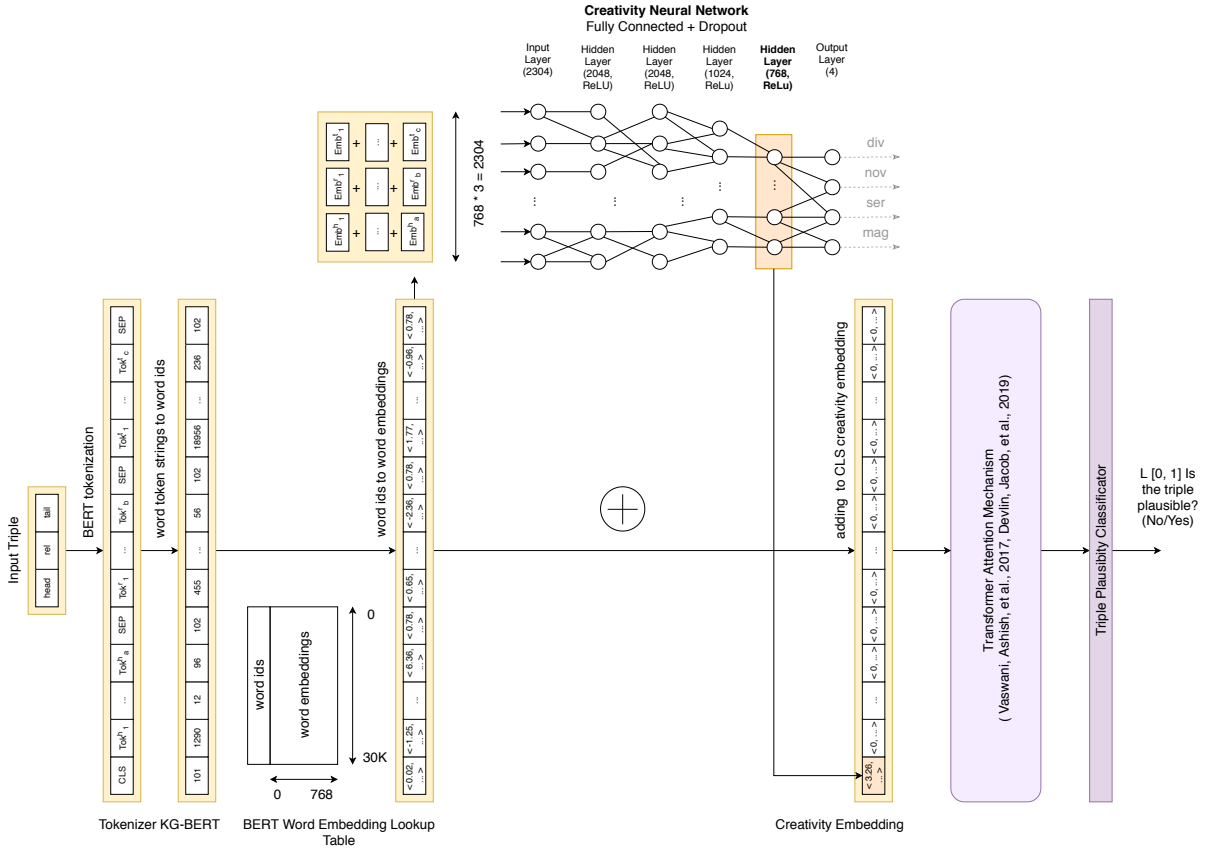

Figure 3: For each triple, Creativity Embedding computed by Creativity Neural Network is added to BERT CLS embedding, defining the Creativity CLS Embedding. A linear classifier on top perform the triple plausibility classification.

relation and tail words in three different sentences. The corresponding token identifiers and embeddings are retrieved through two lookup tables, provided by the BERT model. At the top of Figure 3, we show our creativity neural network. A compact and fixed-size version of the embeddings is obtained from BERT, summing the embeddings of each component of the triple. This compact version feeds the proposed neural network in charge of predicting creativity's four values and producing creativity embedding. The neural network consists of an input layer ($768 * 3$ neurons), an output layer (4 neurons), 4 fully connected hidden layers with a dropout probability = 0.5. The activation function used is $ReLU$. This neural network structure is basic since its main task is to have a flexible last hidden layer adaptable to the technology that would leverage the creativity embedding. The CLS token is one of the most representative tokens to perform classification and other types of predictions. Came to us exploiting CLS token to adding creative embedding of the triple, providing the model with a non-empty CLS, Creativity CLS Embedding. In this case, the penultimate layer has been described with several neurons equal to 768, the same size as the BERT embeddings. On the top of the architecture, a linear classifier is in charge of predictions of the plausibility task relying on Creativity CLS Embedding.

4 Experiment

In this experiment we random sample triples from WordNet11 (Miller, 1995) dataset (50000 train, 5000 validation, 3000 test, with positive and negative labels balanced).

Creativity Neural Network. As stated in the previous sections, we compute the four metrics on each triple dataset to create the ground truth. As a similarity function we use cosine similarity, that returns a value between 0 and 1, with high value for high similarity. We applied the cosine similarity function after transforming words and sentences in embeddings, provided by BERT

model. We encountered slowdowns only with novelty metric. The number of nodes is not predictable a priori in our setting, and the mathematical nature of the formula is sensitive to a high number of nodes. Peaks of memory allocation could occur, as well as long computation time. We limit the failure due to out of memory or timeout of the scheduled jobs applying the "divide et impera" paradigm and other adjustments. The length of the path *p*, seen as recursion deep, is fixed to 5. For each node interested by recursion, the number of maximum neighbor nodes *n* considered is fixed to 20. Once we obtain all the metrics values, we can train the Creativity Neural Network, as a regression problem. We use: as loss criterion mean squared error loss; as optimizer AdamW with learning rate = 0.001, betas = $(0.9, 0.999)$, epsilon = $1e^{-08}$, weight decay = 0.01; as scheduler StepLR with parameters step size = 10 and gamma = 0.1; we train the model for 10 epochs, size batch of 512. To evaluate performance on test set we compute explained variance score = -0.4493, mean absolute error = 0.1733 , mean squared error = 0.0388 and R2 score = -6.7694. Although small values of mean squared and absolute error, R2 tells us that the model do not approximate the distribution better than the "best-fit" line. This is probably due to low entropy of the inputted metrics values, that inspected, result in stationing around 0.5 value.

Triple Plausibility Task. The tokenized triple is inputted to the Creativity Neural Network, obtaining the creativity embeddings. This is added to the CLS embedding token, and the triple flows through the Transformers stack. Therefore, the BERT model is used to make predictions and address the triple plausibility task, putting a linear classifier on top of the Transformer stack. We use as loss function the binary cross-entropy loss function. The literature suggests few epochs and samples for the finetuning process. We finetune BERT for 2 epochs; after we freeze the weights of the model, training only the classifier layer for 3 epochs. We select BERT base uncased as baseline model; as optimizer AdamW with learning rate = $5e^{-05}$, as scheduler a linear scheduler with warm up proportion = 10%; for the classifier dropout probability = 0.5. We fix the maximum sequence length at 100 tokens, as all the triples after tokenization do not exceed this number of tokens.

5 Result and Conclusion

In this paper we investigate if defined creativity embedding improves triple plausibility task, exploiting BERT model. We do not detect an increase in the performance (Table 1), comparing ourselves to KG-BERT results. In this comparison we should point out that the sample used is one fifth of the complete WN11 dataset. This result is somewhat contrary to our expectations, as the creativity embeddings represent in some way a priori information. A possible explanation might be the learning methodology of the creativity embedding: we suppose that a significant loss of information in the process has occurred. Further research might explore other types of embeddings (Grohe, 2020), as graph2vec, and different integration of the proposed metrics. Future experimental investigations may try different parameter configurations. For example, the number of nodes considered intuitively could change the values of metrics as a novelty. Nevertheless, more in-depth data analysis on the used dataset, corresponding knowledge graph, and data correlations could provide additional insights. In future work, we will consider different combinations of metrics defined to train the creativity neural network. It is possible that there are metrics more or not relevant for the task. Selecting metrics strictly relevant will result in a lightening of the computational effort and will give us information about correlations between metrics and results. To conclude, we aim to bring the NLP community's attention to new research topics on creativity.

Acknowledgments

Computational resources provided by HPC@POLITO, which is a project of Academic Computing within the Department of Control and Computer Engineering at the Politecnico di Torino[2]. We thank the reviewers from CLiC-it 2020 conference for the comments and advices.

References

Tim Berners-Lee, James Hendler, and Ora Lassila. 2001. The semantic web. *Scientific american*, 284(5):34–43.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of

[2]http://www.hpc.polito.it

	Number of triples			Model Metrics			
	Train	Val	Test	Accuracy	Recall	Precision	F1
CE+BERT	50000	3000	5000	0.5093	0.8510	0.5102	0.6379
KG-BERT	225162	5218	21088	0.9334	0.9345	0.9324	0.9334

Table 1: Triple plausibility experiment results.

deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Martin Grohe. 2020. Word2vec, node2vec, graph2vec, x2vec: Towards a theory of vector embeddings of structured data. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, PODS'20, page 1–16, New York, NY, USA. Association for Computing Machinery.

P. Karampiperis, A. Koukourikos, and E. Koliopoulou. 2014. Towards machines for measuring creativity: The use of computational tools in storytelling activities. In *2014 IEEE 14th International Conference on Advanced Learning Technologies*, pages 508–512.

Polina Kuznetsova, Jianfu Chen, and Yejin Choi. 2013. Understanding and quantifying creativity in lexical composition. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1246–1258, Seattle, Washington, USA, October. Association for Computational Linguistics.

Carolyn Lamb, Daniel G. Brown, and Charles L. A. Clarke. 2018. Evaluating computational creativity: An interdisciplinary tutorial. *ACM Comput. Surv.*, 51(2), February.

Agnieszka Ławrynowicz. 2020. Creative ai: A new avenue for the semantic web? *Semantic Web*, pages 69–78.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Diego Monti, Enrico Palumbo, Giuseppe Rizzo, and Maurizio Morisio. 2019. Sequeval: An offline evaluation framework for sequence-based recommender systems. *Information*, 10(5):174.

Ulrike Padó, Matthew W Crocker, and Frank Keller. 2009. A probabilistic model of semantic plausibility in sentence processing. *Cognitive Science*, 33(5):794–838.

Marco A.F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko. 2014. A review of novelty detection. *Signal Processing*, 99:215 – 249.

Yu-Ping Ruan, Zhen-Hua Ling, Xiaodan Zhu, Quan Liu, and Jia-Chen Gu. 2020. Generating diverse conversation responses by creating and ranking multiple candidates. *Computer Speech Language*, 62:101071.

Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. 2008. Learning to rank answers on large online qa collections. In *Proceedings of ACL-08: HLT*, pages 719–727.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85, September.

Quan Wang, Bin Wang, and Li Guo. 2015. Knowledge base completion using embeddings and rules. IJCAI'15, page 1859–1865. AAAI Press.

Su Wang, Greg Durrett, and Katrin Erk. 2018. Modeling semantic plausibility by injecting world knowledge. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 303–308, New Orleans, Louisiana, June. Association for Computational Linguistics.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*.