A cognitive inspired method for assessing novelty of short-text ideas

Simona Doboli
Computer Science Department
Hofstra University
Hempstead, USA
simona.doboli@hofstra.edu

Jared Kenworthy
Psychology Department
University of Texas Arlington
Arlington, USA
kenworthy@uta.edu

Paul Paulus
Psychology Department
University of Texas Arlington
Arlington, USA
paulus@uta.edu

Ali Minai
Department of Electrical and Computer Science
University of Cincinnati
Cincinnati, USA
Ali.Minai@uc.edu

Alex Doboli
Electrical and Computer Engineering Department
Stony Brook University
Stony Brook, USA
alex.doboli@stonybrook.edu

***Abstract*—In creativity research a typical problem is that of assessing the novelty of ideas or solutions generated by many people to open ended problems. For datasets larger than a few hundreds, human assessment of novelty becomes time consuming and error prone. Existing novelty detection methods such as: distance based text similarity or language model approaches do not work well for small datasets. Moreover, when compared to human novelty ratings, these approaches fail to capture the same cognitive processes or biases. We are proposing a novel cognitive model inspired by a leaky accumulator decision making models for detecting novel ideas from short text. The model is applied on a collection of ideas generated in a group brainstorming experiment. It evaluates an idea term by term and it accumulates surprise and relevance. The final novelty decision is taken at the end of each idea by means of a threshold. An important component of the model is a small domain dataset which is used to evaluate the surprise of a term's context compared to common domain knowledge. The model is compared with other methods: feature based classifiers, tf-idf similarity distance, and pretrained language models (ULMFIT).**

***Index Terms*—novelty detection, cognitive model, decision making**

I. Introduction

Novelty detection from large amount of text is a known problem with applications in first story detection (i.e. select novel and relevant sentences from a stream of news data) [1], text summarization [2], extraction of novel scholarly papers or abstracts [3]. Our problem is that of detecting novel ideas from a set of solutions generated to an open ended problem. This situation is very frequent in creativity research as well as in real scenarios such as crowdsourcing brainstorming. In both cases, human rating is very tedious, time consuming as well as error prone. For example, novelty rating involves reading the whole data set and then, rating each idea based on its similarity with all other ideas. Large data sets are impossible to keep in one's memory and thus the accuracy of this type of coding decreases with the size of the data. There are currently no automated solutions for any phase of this process, which typically lasts one or two years. Existing automated environments address mainly data collection but have no support for data analysis and modeling [4]. Thus, our problem is to develop a computational method to assess whether an idea is novel in the context of a data set that produces similar results to human assessments. The main challenges of our problem are: (1) detecting novelty in a small collection of short-text from one topic in an unsupervised way, and (2) unreliable human assessments. Our problem is different than that of detecting novelty in other situations mostly because of the nature of the dataset. In our case, all ideas pertain to the same problem - i.e. they share the same topic vocabulary, but they rarely overlap in semantics and the length of each idea is short - 1 to 10 sentences. While in news stream, there are more differences in topics - i.e. vocabulary than semantics and also named entities play a significant role in novelty detection. A typical dataset used in novelty detection is TREC 2002-2004. It consists of news grouped in around 50 topics, with each sentences in each topic marked as relevant or novel [5]. The closest problem and dataset to ours is the one described in Walter and Beck [6] which comes from a crowdsourcing innovation experiment, but with tens of thousands of ideas from multiple contests. The authors point out the difficulty in matching the human assessment of novelty. Their method is based on clustering each idea's tf-idf vector and labeling as high-novelty all ideas from clusters with less than three documents in them. Their best result was an F1 value of 0.63.

II. Related Work

Many novelty detection algorithms are based on approaches from information retrieval [7] which can easily be repurposed to measure similarity between two texts, which then can be used to detect novel texts as those most dissimilar from others. There are word-count methods such as cosine similarity [8], [9], tf-idf methods [1], language model methods based on Kullback-Leibler (KL) divergence [8], [10], information-

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

theoretic measures [11], graph-based methods [3], [12], k-means clustering [6], or Latent Dirichlet Analysis (LDA) [13]. Most of these algorithms use the TREC novelty detection dataset and the best algorithms achieve an F1 score of 0.60-0.75 with precision and recall rates around 65-75%. A comparison study between different methods applied to streams of news story was done in Verheij et al. 2012 [14]. The best method among the ones tested was KL divergence with linear interpolation. Other novelty detection approaches use classification algorithms such as one-class support vector machine (SVM) [15] or neural networks (NN) [16]. Clustering algorithms such as k-means have been also used for novelty assessment [6].

A novelty detection method was proposed by our group Mei et al. (2018) [13] using the same dataset as the one used in this paper. It encodes ideas in a reduced semantic space using LDA and it either reconstructs the original vector using a neural network autoencoder or clusters the LDA vectors to find similar ideas. They observed that neither pairwise similarity nor reconstruction error from the autoencoder were reliable estimates of human assessment of novelty. A clustering approach showed correlation between cluster size and average novelty. The main issues with these term based methods for detecting similarity of short text is that the similarity is at a deep semantic level, not surface word level and that is not captured in the original term vector space. Also, the bag-of-words assumption that the word order does not matter makes it even harder to detect semantic similarity.

Recent deep learning methods looked at encoding semantic similarity between words such as distributed word embeddings: word2vec [17], gloVe [18], fastText [19], or ELMO contextualized word embeddings [20]. These embeddings are obtained by training on very large amounts of data - whole Wikipedia or Google News datasets. They can be used to fine tune neural network language models on smaller datasets. Initial attempts composed these word embeddings to produce longer text embeddings [21]–[23]. Approaches vary from simple averaging [21], deep averaging [23] to smooth-inverse averaging [24] of word embeddings. Most of these average based sentence representations have been used for training on supervised tasks such as sentence classification, paraphrase similarity, or entailment. But word embeddings tend to capture well the most common sense of a word, but not others. Thus, in general, an average of word embeddings without further training for semantic similarity does not lead to good semantic encoding of short text [21]. Another straight-forward approach to generating sentence level embeddings is through the use of RNN's either with LSTM [25] or GRU units [26] . Distributed word embeddings from a sentence are fed sequentially into a unidirectional or bidirectional RNN, with the final RNN state becoming the sentence level embedding [27]. Bidirectional LSTM models tend to outperform unidirectional ones [27] due to their encoding of context coming from both ends of the sentence. Learning in these models is either in the form of a language model - predict the next word in a sequence or supervised global sentence classification. Word embeddings are initialized with pre-trained embeddings (e.g. word2vec, GloVe, fasttext, etc.) and fixed or tuned during learning or trained from scratch. Adding structure to a linear RNN improved performance: Socher et al. 2013 [28] trained a recursive neural network model on sentiment classification. Each sentence was parsed into a binary tree and each node corresponding to a word was represented as a vector [28]. Composition of word vectors was then used as input in a classifier. The model employed for training a parsed dataset, with all sub-phrases labeled with a fine grained sentiment. A similar approach was used in [27] by using an LSTM tree network over a constituency or dependency parser.

Task independent or universal sentence embeddings that could be used as is or fine tuned for subsequent tasks or data sets have been the focus of recent research. Inspired by the transfer learning abilities of deep learning methods for image recognition, a number of approaches have been proposed for natural language processing (NLP). Conneau et al. (2018) [29] aimed at learning universal sentence encoders using different NN architectures. The best performing model on both the original training task and the transfer tasks was the bidirectional LSTM with max pooling, especially on semantic relatedness tasks. Other approaches to text embedding for transfer learning include ULMFIT [30], ELMO [20], BERT [31], universal sentence encoders [32], skip-thought vectors [33]. Skip-thought vectors [33] were among the first universal sentence encoders. The model is an encoder-decoder architecture build as RNN with GRUs with the objective of predicting the next and previous sentences given a sentence. The model performs at a similar level with models trained on a particular data set only. It can be easily used on sentence similarity tasks. ULMFIT is a 3 layer LSTM network trained as a language model on a large general domain corpus [30]. The model then can be first fine-tuned on domain data as a language model, then further trained an outside classifier. The general pre-training phase helps especially in case of small data sets. The performance of the pre-trained model with fine tuning approaches that of a fully trained model, but with 10 to 20 times less data. The ELMO model [20] has at its core a 2 layer bidirectional LSTM network and its goal is to produce context-dependent word embeddings as a weighted sum of vectors along all layers in the model. The model is trained as a language model on a very large corpus and its context dependent vectors can be used as is or with fine tuning in any downstream task. Both BERT [31] and universal sentence encoder (USE) methods [32] are based on the transformer architecture [34] which is an attention based NN with no recurrence, but with a deep architecture - over 6 layers. BERT is a deep bidirectional encoder transformer model, while the universal sentence encode uses either the transformer (USE-T) or the deep averaging network (USE-D) [23]. BERT is trained to encode both single sentence and double sentences and it uses masked input and next sentence prediction during pre-training. It is trained with the BooksCorpus and Wikipedia and it can be fine tuned on the target dataset and task. USE models generate a fixed length representation for a sentence

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

input. USE models use also a combination of unsupervised learning with Wikipedia data and supervised learning using the Stanford Natural Language Inference (SNLI) corpus [35].

The only deep learning method for detecting novel documents was proposed by Ghosal et al. (2018) [36]. The method is based on InferSent, the sentence encoder proposed by Conneau et al. (2018) [29] and the convolutional neural network (CNN) sentence encoder proposed by Kim (2014) [37]. A source document is compared with a set of target documents: each sentence in the target document is mapped onto the most similar sentence in the source document using the InferSent BiLSTM (bidirectional long-short term memory) sentence encoder [29]. A combined vector is then generated for each target sentence, and concatenated to generate a relative document representation. This is then fed through a CNN network and a softmax classification into redundant or not classes. The results obtained by this method improve on previous probabilistic language model and cosine similarity as well as paragraph vector methods and just InferSent methods, especially on precision measure. An interesting outcome is that results are highly dependent on the dataset.

While deep learning methods allow a better representation of semantic meaning at the level of words or even sentences, they do require a large amount of data for in domain training. Also, the method in Ghosal et al. (2018) uses supervised training, which is not feasible in our situation. In this paper we aim to develop a method inspired by cognitive processes of decision making and novelty detection and that is able to explain and replicate human novelty assessment of ideas or solutions on open-ended problems. We do not intend to develop a general purpose text novelty detection method. This is the reason we only use the one dataset of ideas we have obtained experimentally at this time.

Our method is based on a repeated, incremental process that assesses the surprise and relevance of each word in an idea and the novelty of its context. This approach attempts to emulate the human cognitive process of novelty assessment - based on reading the whole dataset and previous domain knowledge. The decision making aspect - whether an idea is novel or not - is made using a cognitive accumulator based model [38]–[40]. Since all ideas come from the same domain, we use a small database of domain knowledge to augment the perception of relevance and context surprise of various terms in an idea. Results applied to our ideas database show that the average estimated surprise values increase with human assessed novelty while average estimated relevance values decrease. We also show that the use of the external domain dataset has a positive influence over novelty assessment. We compare our method with three other approaches: feature-based classifiers, cosine similarity of tf-idf vector space representations and a transfer learning LSTM network trained as a language model called ULMFIT [30]. Our novel cognitive model obtained the best F1-score (0.578) among all methods tested. While the match with human assessment of novelty is not great, the main purpose of our method is to model the cognitive process of novelty assessment. We believe that such an approach can be used in the future to test various biases in human assessment of novelty.

III. Motivation

The main motivation for the proposed method is to model aspects of the human cognitive process being used in assessing novelty of ideas generated situations or problems that require a creative solution. While novelty is easy to define - e.g. a unique solution - it is actually very difficult to explain or derive computationally. There are no correlates of human assigned novelty to any surface level text features including: number of unique words, number of keywords, text length, and so on (see Section V). Also, inter-rater correlations of novelty are also usually very small - i.e. raters tend to differ in their rating of novelty for the same dataset We are interested in deriving a computational method of novelty assessment that models the human raters cognitive process.

Humans do not rely on precise statistical estimation of probabilities of words or sentences or ideas [41]. We are making the assumption that they are using an incremental process similar to a sequential decision making. While reading an idea word by word, each word or expression and its local context trigger small changes in their current assessment of novelty. If the word seems new in the context of the dataset or the domain (e.g. 'pyramid run'), or if the word is more frequent, but in a novel local context (e.g. 'roller-skate soccer') it will increment the current assessment of novelty. The final novelty assessment is an aggregate over many intermediary assessments that occur while an idea is being read and processed. Our proposed model is informed by cognitive models of decision making as well as novelty signals in the brain.

For example, in the accumulator model of decision making each possible outcome has its own accumulator that integrates noisy evidence over time. The moment one of these accumulators has reached its threshold a decision is made. This model assumes that all accumulators are independent. The drift diffusion model [38], [39] relaxes this assumption. In its basic form it consists of a single accumulator with two decision boundaries. Evidence accumulates at a drift rate under noisy conditions and when one of the decision boundaries is crossed a decision is made. The leaky competing accumulator model [40] extends the basic drift-diffusion model with separate accumulators for each choice. They accumulate evidence over time under leaky conditions and compete against each other via lateral inhibition. A decision is made when one of the accumulators crosses its threshold.

The method we propose here is based on modeling novelty assessment of ideas as a binary decision model. Evidence is accumulated over time as words from an idea are processed one by one. Each word and its context triggers a local surprise and relevance estimation that is accumulated over time. The overall evidence at the end of the text is used to decide whether an idea is novel or not. Neural signals that encode the novelty of a stimulus have been discovered in the brain, for example in the substantia nigra/ventral tegmental area (SN/VTA) [42].

Hippocampus, on the other side, seems to be most active when a stimulus and its context disagree, also called novelty association [43].

IV. Model description

The structure of the cognitive model is shown in Fig. 1. It consists of a surprise component and a relevance component. An idea is fed term by term into both components. The total surprise of a term is composed of the surprise of the term in the idea dataset and the surprise of the term in its surrounding context compared to commonly encountered contexts from the external domain dataset. The relevance of the term is evaluated against the domain dataset. Both total surprise and relevance are accumulated individually over all terms in an idea. The decision of low or high novelty is taken based on the accumulated relevance and surprise values.

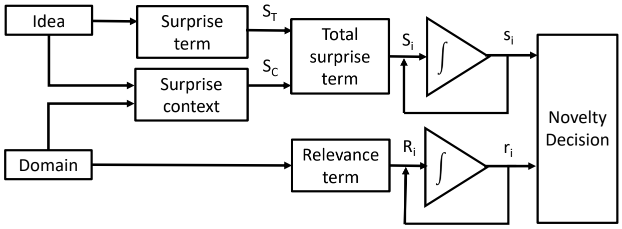

Fig. 1. Structure of the cognitive novelty assessment model

The input into the model is the collection of short text denoted by T with N_T short-texts: $T = \{T_1, T_2, \dots T_{N_T}\}$. The set of terms in T defines the dictionary of the text collection $Dict_T$. A term is defined by the lemmatized form of a word and its part of speech. The second input is a collection of domain relevant text (D) with N_D texts: $D = \{D_1, D_2, \dots D_{N_D}\}$ and its own dictionary of terms: $Dict_D$. The assumptions about the domain collection D are: (1) it is a representative set of texts describing the major topics of the domain, and (2) there is no text outside the domain (e.g. no non-relevant text).

An idea from the collection T_k is composed of a sequence of p terms: $T_k = \{t_1, t_2, ...t_i, ..., t_p\}$. For each term t_i in T_k, its surprise ($S_(t_i)$) and relevance ($R_(t_i)$ are computed as follows:

1) The surprise of the term $S_T(t_i)$ is given by the normalized document term frequency in the idea collection (T), $DF_T(t_i) = |\{T_m, t_i \in T_m, \forall T_m \in T\}|/N_T$:

$$S_T(t_i) = 1 - DF_T(t_i) \tag{1}$$

2) The surprise of t_i's context is denoted by the terms that appear around it in the idea T_k but not around it in the domain collection D ($S_C(t_i)$). If a term is not in the collection dictionary, then its surprise context is the maximum value 1.

$$S_C(t_i) = \frac{|C_T(t_i) \setminus C_D(t_i|}{c_{size}} \tag{2}$$

where $C_T(t_i)$ is the set of terms within a window of size c_{size} around t_i :

$$C_T(t_i) = \{t_j |\, |t_j - t_i| < c_{size}, t_j \neq t_i, t_j, t_i \in T_k\} \tag{3}$$

and $C_D(t_i)$ is the set of terms within a window of size c_{size} around all occurrences of t_i in the domain collection D:

$$C_D(t_i) = \{t_s |\, |t_s - t_i| < c_{size}, t_s \neq t_i, t_s, \forall t_i \in D\} \tag{4}$$

3) The total surprise of t_i is the product of it's surprise in the idea collection T and the surprise of its context compared to the domain dataset D: $S(t_i) = S_T(t_i) \cdot S_C(t_i)$
4) The relevance of term t_i is evaluated as normalized document term frequency in the domain collection D, $DF_D(t_i) = |\{D_m, t_i \in D_m, \forall D_m \in D\}|/(N_D + 1)$:

$$R(t_i) = \begin{cases} DF_D(t_i) & \text{if } t_i \in Dict_D, \\ \dfrac{1}{N_D + 1} & \text{if } t_i \notin Dict_D. \end{cases} \tag{5}$$

The surprise and relevance of all terms in a text T_k are accumulated to produce the novelty decision. We assume that accumulation of high levels of surprise leads to a high novelty decision, while accumulation of high levels of relevance in the presence of low surprise leads to a decision of low novelty. For the present model we consider that accumulation is done using two independent discrete leaky accumulators one for surprise ($s(i)$) and one relevance ($s(i)$) as shown below:

$$\begin{aligned} s(i) &= \frac{1}{|Tk|}(\lambda_S \cdot s(i-1) + k_S \cdot S(t_i) + \eta_S(i)) \qquad (6) \\ r(i) &= \frac{1}{|Tk|}(\lambda_R \cdot r(i-1) + k_R \cdot R(t_i) + \eta_R(i)) \end{aligned}$$

with i the time step when t_i is processed, $|T_k|$ is the length of the evaluated text, $\lambda_{(.)}$ are the leaky coefficients, $\eta_{(.)}(i)$ the noise terms, k_S is the surprise coefficient, k_R the relevance coefficient. It is assumed that the initial conditions are reset to 0 at the beginning of each text : $s(0) = 0$ and $r(0) = 0$. This can be relaxed to check if the ratings of previous ideas affect future ratings. We found that similar ratings to consecutive ideas tend to appear in bursts, which may be due to people generating bursts of either high novelty or low novelty ideas or to human assessment of consecutive ideas.

The novelty decision is taken as follows:

$$n(T_k) = \begin{cases} high & \text{if } s > \theta_S \text{ and } r > \theta_R, \\ low & \text{otherwise.} \end{cases} \tag{7}$$

with $\theta_s(t)$ the threshold for high surprise and $\theta_r(t)$ the threshold for minimum relevance. These thresholds can be constant or can vary over time with previous decisions. In the present model we considered them fixed.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

V. Data description

The ideas dataset T used in this paper was generated during group behavioral experiments conducted at University of Texas at Arlington [44]. Participants in 57 groups of four were asked to generate ideas on a new sport. The data set contains $N_T = 1480$ ideas. The posts length range from 1 to 428 words, with an average of 38 words per post. Experts rated all ideas with a novelty value between 1 (low-novelty) to 5 (high-novelty) compared to other posts. All posts were read before being rated.

The domain collection of sports D was collected from Wikipedia and it has $N_D = 1144$ sports with title and a short description. Descriptions varied in length from 0 to 950 words with an average of 190 words. The data was collected from a recent Wikimedia dump [45].

Currently, this is the only brainstorming ideas dataset that we have processed. In the future we plan on testing our model on other datasets as they become available.

A. Data Preprocessing

All ideas from T were preprocessed with the following steps: tokenize each idea into sentences and into words, eliminate special characters, spell check for errors, extract the parts of speech for each word using Stanford parts of speech tagger, eliminate stop words using Porter's stop words corpus, keep only nouns and verbs, lemmatize words using Wordnet Lemmatizer. From all remaining words, only the ones that appear at least two times were kept. The dictionary $Dict_T$ had 2,395 words. At the end of this process 6 ideas had no remaining words and were not included in the results.

All sports from D were preprocessed in the same way as T ideas. The dictionary containing all remaining terms in the sports descriptions and all terms in the sports titles ($Dict_D$) had 11,223 words.

VI. Results

The cognitive model was tested on the data set described above. The original novelty values from 1 to 5 were converted into $high = 1$ and $low = 0$: values 4 and 5 were considered $high$, while 1 to 3 low. The dataset is approximately balanced with 756 $high$ novelty ideas and 717 low novelty ideas.

The first experiment has the following parameter values: $k_S = k_R = 1$, non-leaky ($\lambda_S = \lambda_R = 0$), no-noise ($\eta_R = \eta_S = 0$) and context size $c_{size} = 2$. The mean surprise and relevance in each $high$ and low novelty class of ideas is shown in Table I. The average surprise increases for $high$ novelty ideas, while the average relevance decrease. Both surprise and relevance are in the 0 to 1 interval.

TABLE I
Average surprise and relevance (Experiment 1)

Novelty	**Average surprise**	**Average relevance**
low	0.57	0.16
$high$	0.65	0.13

We varied the surprise and relevance thresholds (θ_S and θ_R) to find the values that minimize the number of missclassfied ideas. The θ_S values were varied from 0.1 to 0.9 in increments of 0.1, while the θ_R values were varied from 0 to 0.4 in increments of 0.05. The best values were $\theta_S = 0.6$ and $\theta_R = 0$. The largest number of correctly classified ideas was 846 or 57.39% from the total of 1474 ideas. The $F1$ score was 0.572. Interestingly, it seems that the relevance factor did not improve novelty results.

Table II shows F1-scores obtained with different values of leaky coefficients $\lambda_{(.)}$ and noise standard deviations $std_{(.)}$. The mean of the noise is set to 0 in all cases. The table shows the best values for $theta_S$ and $theta_R$ (i.e. that maximize the number of correctly classified ideas). The best relevance threshold is again 0 independent of the condition. It can be seen that the results are relatively robust to noise as adding some does not change the results significantly. Added leak in relevance and surprise (rows 4 to 8) increases the overall $F1$ score. This is interesting as the non-zero leak condition models better the cognitive decision making process.

TABLE II
Summary results

#	λ_S	λ_R	std_S	std_R	**F1 score**	θ_S	θ_R
1	0.0	0.0	0.0	0.0	0.572	0.6	0.0
2	0.0	0.0	0.01	0.001	0.573	0.6	0.0
3	0.0	0.0	0.1	0.01	0.564	0.6	0.0
4	0.012	0.0002	0.001	0.0001	0.577	0.7	0.0
5	0.012	0.0002	0.0	0.0	0.578	0.7	0.0
6	0.02	0.001	0.001	0.0001	0.574	0.7	0.0
7	0.02	0.001	0.01	0.001	0.573	0.7	0.0
8	0.05	0.01	0.01	0.001	0.558	0.7	0.0

Looking in more detail at the precision and recall in each novelty category, for the same conditions shown in Table II, it can be seen that some cases lead to a better precision for low novelty ideas while others for $high$ novelty ideas. Table III shows the precision and recall in both novelty categories. The row index in Table III corresponds to the row index from Table II. No leak cases (rows 1 to 3) have higher precision for the $high$ novelty class, while leak cases (rows 4-7) lead to higher precision in the low novelty class. The only exception is row 8, corresponding to high values of $\lambda_{(.)}$ coefficients with under 50% precision for the low novelty class. Since relevance does not improve classification accuracy, we ran an additional experiment to see the effect of surprise context of

TABLE III
Precision-recall in low/high category

#	P_{high}	R_{high}	P_{low}	R_{low}
1	0.605	0.581	0.539	0.565
2	0.600	0.582	0.546	0.564
3	0.566	0.577	0.563	0.551
4	0.513	0.605	0.647	0.557
5	0.513	0.606	0.648	0.558
6	0.533	0.596	0.619	0.557
7	0.526	0.596	0.624	0.555
8	0.621	0.566	0.497	0.555

a term ($S_C(t_i)$) in the overall novelty assessment. Relevance and surprise context are the only two terms that depend on the D domain collection. We set $S_C(t_i) = 1$ in no leak and no noise conditions. In this case the surprise term depends only on the T collection. The smallest number of missclassified ideas was 641 for $\theta_S = 0.95$ and $\theta_R = 0$ with an $F1$ value was 0.564. This is smaller than the same case with surprise context (row 1 in Table II) which had an $F1$ score equal to 0.572. This shows that the surprise of the context using information from the additional domain collection D improves novelty classification.

In another experiment, we kept the same parameter values used in row 5 Table II and III, but with surprise and relevance values not reset to 0 at the beginning of each idea, but to 50% of the values from the previous idea. This means that we introduced a cognitive inertia in the model in both surprise and relevance from one idea to the next. The smallest number of misslassified ideas was 633, with a $F1$ score of 0.568, obtained for $\theta_S = 0.9$ and $\theta_R = 0$. While these results are slightly worse than with no cognitive inertia (see row 5 in Table II and in Table III), it is interesting that the the model improved the precision and recall for $high$ novelty ideas, but lowered them for low novelty ideas.

A. Comparison with classifiers

We extracted from the T collection for each T_k fifteen features as follows: (1) the number of most common words defined as appearing at least 50 times in the data set, (2) the number of least common words defined as with a number of occurrences less than 10, (3) the number of other words, (4) number of words that appear in a sports title in the D collection, (5) number of words that appear in a sports equipment list - extracted from Wordnet [46], (6) number of words in the keywords set from D obtained using the TextRank method from gensim [47], (7) number of frequent bigrams from the list of bigrams with a frequency of at least 15, (8) number of frequent trigrams from the list of trigrams with a frequency of at least 7, (9) number of sentences in T_k, (10) number of original words before preprocessing, (11) number of words after stemming and removal of stop words, (12) number of unique words, (13) length of the largest sentence, (14) length of the smallest sentence, and (15) average sentence length.

The dataset was normalized and split into 67% training and 33% testing. We trained a large number of classifiers including: support vector machines (SVM), neural networks (NN), logistic regression, decision trees, k nearest neighbor (kNN), and naiive Bayes. The best results were obtained by a two layer NN and kNN with F1 scores on the test set ranging from 0.54 to 0.58. The scores varied significantly with the testing set which means that the classifiers were overfitting the training data. Compared to the results of the cognitive model, these classifier results are more variable and in most cases with a lower performance.

B. Comparison with $tf-idf$ vector space method

Using vector space methods, where each T_k idea is expressed as a vector of $tf-idf$ values of the same size as the dictionary $Dict_T$, we computed the pairwise cosine distance for each T_k. A raw novelty between 0 and 1 was computed as $raw_{nov}(t_i) = 1 - max\{cos_{dist}(t_i, t_j), \forall t_j, t_j \neq t_i\}$. Novelty was considered $high$ if raw_{nov} was higher than a threshold and low otherwise. The smallest number of missclassified ideas was obtained for a threshold equal to 0.85, with a best $F1$ score of 0.35. The novelty based on cosine similarity assigned almost all ideas a low novelty, only 17 ideas were deemed $high$ novelty. These results are much worse than the ones obtained with the classifiers and the proposed cognitive novelty method. This shows that this vector space method in the original dictionary space does not work well for small data sets.

C. Comparison with ULMFIT pretrained model

The ULMFIT model is a pretrained language model using a three layer AWD-LSTM [48] recurrent network architecture [30]. The model has 400 size trained embeddings for each word in a 60K vocabulary. The model was trained on Wikipedia dataset as a language model - predict the next word. The input layer is an embedding layer with 400 nodes and the output layer has also 400 nodes - the predicted embedding of next word projected onto a softmax layer over the whole vocabulary. One of the main features of the ULMFIT model is that it can be fine-tuned on a new dataset and it allows training of out of original vocabulary terms. It achieves similar results but with much less data than a model trained from scratch. We believe that ULMFIT is the closest to our cognitive model in measuring the surprise of an idea. We have fine-tuned the pretrained ULMFIT on our idea and/or domain dataset(s) as a language model to predict the next word. We use perplexity to compute the surprise of an idea. Perplexity is a measure of how well the model predicts all the words in an idea. The expectation is that high novelty ideas will have a higher perplexity than low novelty ideas, since low novelty ideas will appear more often, hence have a lower prediction error. We used the following expression to compute perplexity:

$$P_r(T_k) = 2^{-(1/|T_k|)\cdot\sum_{i=1}^{i=|T_k|} log(Prob(t_i))} \quad (8)$$

where $Prob(t_i)$ is the softmax probability of the true term t_i and $|T_k|$ the length of the idea.

We use perplexity values obtained for each idea to assign high and low novelty values. Ideas with perplexity higher than a threshold will be assign high novelty, while lower than a threshold, low novelty. We compute the Area Under the Receiver Operating Characteristic Curve (AUROC) score which varies the threshold level, with a higher score indicating a higher true positive rate than false positive rate.

We have conducted several fine-tuning experiments with the results shown in Table IV: (a) no fine tuning, (b) fine tuning on ideas, (c) fine tuning on the sports domain data, (d) fine tuning on both sports domain and ideas datasets, (e) AWD-LSTM trained on ideas only with random weights and

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

embeddings. All AUROC scores are very small, indicating that perplexity fails to predict novelty (e.g. a score of about 0.5 indicates random prediction). Expectedly with more fine-tuning especially on ideas, the model has lower scores because perplexity values go down over all ideas. Interestingly, the 'best' results are obtained for fine-tuning on the sports domain dataset. This shows that there is valuable information in adding domain data.

TABLE IV
AUROC SCORE FOR ULMFIT MODEL

Case	$AUROCscore$
a	0.5332
b	0.5249
c	**0.5354**
d	0.4922
e	0.5351

VII. Discussion

We proposed a novel cognitive novelty assessment method inspired by the leaky-accumulator model of decision-making [38]–[40]. The method views the assessment process as an iterative process, where surprise and relevance are computed as new words are seen and processed in their local context as well as in a domain relevant context. We propose this method for assessing novelty of small size datasets of creative ideas or solutions where probabilistic or term vector space methods do not work well. The main goal of the method is to mimic the cognitive processes that take place when human evaluators assess a large set of ideas on a particular topic or problem.

The method uses in addition to the idea data set a small but relevant domain data set, which in this case was extracted from Wikimedia [45]. The domain collection is used to compute the novelty of the context in which a term appears in an idea versus basic domain knowledge. The context surprise of a term improves novelty assessment results, showing the importance of the additional domain knowledge. The relevance of each term in an idea was also evaluated using the domain dataset. But extensive experimental results showed that this term is not-relevant in obtaining the best novelty assessment. The results are improved slightly when memory of past surprise is added as a leak term and in the presence of some noise. The best results are obtained with some leak and no noise: $F1 = 0.578$. Leak seems to increase the precision of low novelty ideas, while decreasing it slightly for $high$ novelty ideas.

We compared our method with many others, from classifiers and distance based similarity to pretrained language model. The highest $F1$ score was obtained with our model. Among the classifiers, the best results were given by multilayer perceptron networks with two hidden layers and the kNN method - with best results having $F1 = 0.58$, but with a very high variability with respect to the testing data set. The cosine similarity among tf-idf vector space representations of ideas gave the poorest results, with a $F1$ score equal to 0.35. The pretrained ULMFIT model, even though it uses a very large dataset (Wikipedia) for pretraining, it did not produce better novelty predictions using perplexity. We chose this model because it allowed us to emulate the human cognitive process of evaluating surprise with different degrees of knowledge and expertise - depending on which dataset was used for fine-tuning. The model proposed earlier by our group [13] used the same dataset as ours but it did not do full novelty prediction. It used a projection of the term space onto the topic space followed by clustering which showed an inverse relationship between cluster size and novelty.

VIII. Conclusions

In this paper we propose and evaluate a new cognitive model for novelty assessment of creative ideas or solutions expressed as short text. The method was used to evaluate the novelty of a dataset of ideas generated in a group brainstorming experiment [44]. Conceptually, the model emulates the cognitive iterative evaluation processes that take place while one is reading and evaluating an idea, after going through the entire dataset. The method is compared with classic classifiers, tf-idf cosine similarity and a pretrained language model. The cognitive model produced the best F1-score. A key element of the model is an additional domain relevant dataset which is used to compute the surprise of a term's context and its relevance. A major source of creative thinking is the generation of novel conceptual combinations from existing concepts [49], [50]. It is very difficult for machines to discover these novel combinations, unless they have been trained on a large amount of data. The goal of the additional domain dataset is to uncover novel uses of a term compared to common knowledge without using a very large dataset. The surprise of a term's context, does, indeed improve the accuracy of novelty assessment. The main advantage of the proposed model is that the results and parameters have a direct cognitive interpretation and that it can be tuned to match human novelty assessment.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant BCS 1247971. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] M. Karkali, F. Rousseau, A. Ntoulas, and M. Vazirgiannis, "Using temporal IDF for efficient novelty detection in text streams," *CoRR*, vol. abs/1401.1456, 2014.

[2] D. Das and A. F. T. Martins, "A survey on automatic text summarization," Technical Report, 2007.

[3] R. K. Amplayo, S. Hong, and M. Song, "Network-based approach to detect novelty of scholarly literature," *Inf. Sci.*, vol. 422, no. C, pp. 542–557, Jan. 2018.

[4] J. Pennebaker, M. Mehl, and K. Niederhoffer, "Psychological aspects of natural language. use: our words, our selves," *Annu Rev Psychol.*, pp. 547–577, 2003.

[5] I. Soboroff and D. Harman, "Novelty detection: The TREC experience," in *Proc. of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, 2005, pp. 105–112.

[6] T. P. Walter and A. Back, "A text mining approach to evaluate submissions to crowdsourcing contests," in *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, 2013.

[7] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, USA: Cambridge University Press, 2008.

[8] J. Allan, C. Wade, and A. Bolivar, "Retrieval and novelty detection at the sentence level," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, ser. SIGIR '03, 2003, pp. 314–321.

[9] Y. Zhang, F. S. Tsai, and A. T. Kwee, "Multilingual sentence categorization and novelty mining," *Inf. Process. Manage.*, vol. 47, no. 5, pp. 667–675, Sep. 2011.

[10] Y. Zhang, J. Callan, and T. Minka, "Novelty and redundancy detection in adaptive filtering," in *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '02. New York, NY, USA: ACM, 2002, pp. 81–88.

[11] T. Dasgupta and L. Dey, "Automatic scoring for innovativeness of textual ideas," in *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence ¿ Knowledge Extraction from Text*, 2016.

[12] M. Gamon, "Graph-based text representation for novelty detection," *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, pp. 17–24, 2006.

[13] M. Mei, X. Guo, B. C. Williams, S. Doboli, J. B. Kenworthy, P. B. Paulus, and A. A. Minai, "Using semantic clustering and autoencoders for detecting novelty in corpora of short texts," in *Proc. 2018 World Congress on Computational Intelligence (WCCI'18)*, 2018.

[14] A. Verheij, A. Kleijn, F. Frasincar, and F. Hogenboom, "Comparison study for novelty control mechanisms applied to web news stories," *Proc. of the The 2012 IEEE/WIC/ACM Int. Joint Conf. on Web Intelligence and Intelligent Agent Technology - Volume 01*, pp. 431–436, 2012.

[15] T. Tomiyama, K. Karoji, T. Kondo, Y. Kakuta, and T. Takagi, "Meiji university web, novelty and genomics track experiments," in *NIST Special Publication 500–261: The Thirteenth Text REtrieval Conference (TREC 2004)*, 2004, pp. 13–17.

[16] L. Manevitz and M. Yousef, "One-class document classification via neural networks," *Neurocomput.*, vol. 70, no. 7-9, pp. 1466–1481, 2007.

[17] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[18] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[19] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv:1607.04606*, 2016.

[20] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. of the 2018 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, 2018, pp. 2227–2237.

[21] J. Wieting, M. Bansal, K. Gimper, and K. Livescu, "Towards universal paraphrastic sentence embeddings," in *ICLR 2016*, 2016.

[22] M. Yu, M. R. Gormley, and M. Dredze, "Combining word embeddings and feature embeddings for fine-grained relation extraction," in *Proc. of the 2015 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado, 2015, pp. 1374–1379.

[23] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III, "Deep unordered composition rivals syntactic methods for text classification," in *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th Int. Joint Conf. on Natural Language Processing (Volume 1: Long Papers)*, Beijing, China, 2015, pp. 1681–1691.

[24] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *ICLR 2016*, 2016.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[26] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. of 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734.

[27] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," in *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th Int. Joint Conf. on Natural Language Processing (Volume 1: Long Papers)*, Beijing, China, 2015, pp. 1556–1566.

[28] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1631–1642.

[29] A. Conneau and D. Kiela, "Senteval: An evaluation toolkit for universal sentence representations," in *http://arxiv.org/abs/1803.05449*, 2018.

[30] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Vol. 1)*, Melbourne, Australia, 2018, pp. 328–339.

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1*, Minneapolis, Minnesota, 2019, pp. 4171–4186.

[32] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. St. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, and R. Kurzweil, "Universal sentence encoder for English," in *Proc. of 2018 Conf. on Empirical Methods in Natural Language Processing: System Demonstrations*, 2018, pp. 169–174.

[33] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler, "Skip-thought vectors," in *Proc. of the 28th Int. Conf. on Neural Information Processing Systems*, ser. NIPS'15. MIT Press, 2015, pp. 3294–3302.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008.

[35] S. R. Bowman, G. Angeli, C. Potts, , and C. D. Manning, "A large annotated corpus for learning natural lan- guage inference," in *EMNLP 2015*, 2015.

[36] T. Ghosal, V. Edithal, A. Ekbal, P. Bhattacharyya, G. Tsatsaronis, and S. Chivukula, "Novelty goes deep. a deep neural solution to document level novelty detection," in *27th International Conference on Computational Linguistics (COLING 2018), Santa Fe, New Mexico, USA*, 2018, pp. 2802–2813.

[37] Y. Kim, "Convolutional neural networks for sentence classification," in *EMNLP 2014*, 2014.

[38] R. Ratcliff and G. McKoon, "The diffusion decision model: theory and data for two-choice decision tasks," *Neural computation*, vol. 20, pp. 873–922, 2008.

[39] R. Ratcliff, "A theory of memory retrieval," *Psychological Review*, vol. 85, pp. 59–108, 1978.

[40] M. Usher and J. McClelland, "Loss aversion and inhibition in dynamical models of multialternative choice," *Psychol Rev.*, vol. 111(3), pp. 757–769, 2004.

[41] A. Tversky and D. Kahneman, "Judgment under uncertainty: Heuristics and biases," *Science*, vol. 185, no. 4157, pp. 1124–1131, 1974.

[42] N. Bunzeck and E. Düzel, "Absolute coding of stimulus novelty in the human substantia nigra/vta," *Neuron*, vol. 51, no. 3, pp. 369 – 379, 2006.

[43] P. P. Thakral, S. S. Yu, and M. D. Rugg, "The hippocampus is sensitive to the mismatch in novelty between items and their contexts," *Brain research*, vol. 1602, pp. 144–152, 2015.

[44] L. E. Coursey, R. T. Gertner, B. C. Williams, J. B. Kenworthy, P. B. Paulus, and S. Doboli, "Linking the divergent and convergent processes of collaborative creativity: The impact of expertise levels and elaboration processes," *Frontiers in Psychology*, vol. 10, p. 699, 2019.

[45] "Wikimedia downloads," https://dumps.wikimedia.org.

[46] C. Fellbaum, *WordNet. An electronic lexical database*, C. Fellbaum, Ed. MIT, 1998.

[47] R. Mihalcea and P. Tarau, "Textrank: Bringing order into texts," in *the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, 2004.

[48] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and optimizing LSTM language models," in *6th Int. Conf. on Learning Representations, ICLR 2018, Vancouver, BC, Canada*, 2018.

[49] M. Mobley, L. Doares, and M. Mumford, "Process analytic models of creative capacities: Evidence for the combination and reorganization process," *Creativity Research Journal*, vol. 5, pp. 125–155, 1992.

[50] T. M. Amabile and J. S. Mueller, "Studying creativity, its processes, and its antecedents: An exploration of the componential theory of creativity," in *Handbook of organizational creativity*, J. Z. . C. Shalley, Ed. Mahwah, NJ: Lawrence Erlbaum Associates, 2007.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE