**Proposal**

**Introduction**

Creativity is perhaps the most important skill to cultivate in modern day culture. The scale and complexity of problems is so huge that the impact of a solution is even greater (e.g., climate change, inequality, poverty). Creativity is commonly defined as the ability to produce work that is both novel and useful within a social context (Stein, 1953). And it is essential that we better understand how people come up with incredible creative ideas to solve the global problems human currently face, such as climate change.

When researchers study creativity they often ask people to complete divergent thinking tasks, where people are asked to come up with as many ideas as possible surrounding a topic or problem (Silvia et al., 2008). The Alternative Uses Task (Hass, 2017) is perhaps the most commonly used divergent thinking task. It requires participants to list non-obvious uses for a common object (e.g., brick, fork, towel). For example, when the object is brick, a participant could give creative responses such as boat anchor and hammer. De Dreu, et al. (2008, 2011) describe two pathways to coming up with creative ideas, whether on tasks like the AUT or in the real world. One path is the persistent path where a person focuses on a category of responses and digs deep to find a creative solution, for instance during the AUT focusing on creative things you can fixate using a brick such as a bookend, paperweight or boat anchor. The other path is the flexible path, where people use a broad number of cognitive categories to forage for creative ideas, for example by focusing on creative things you can build (e.g., insect house), fixate, or use scraping to achieve (e.g., chalk, nail file) on the AUT brick task. In both pathways, people must efficiently search through their memory to come up with creative ideas.

People seem to search for information similarly to how animals forage for food (Fu & Pirolli, 2007; Hills et al., 2012; Hills et al., 2015; Payne, Duggan, & Neth, 2007). Animals tend to forage for food by moving through different patches of resources, for example a patch of ants or a patch of fruits in a tree. Animals, such as bees, first persist within a patch of, say flowers, and then switch to another patch at an optimal point in time - when the resources in the current patch drop below their average intake rate over the environment (Charnov, 1976). The way people flexibly switch from searching for information within one patch to another in semantic memory follows a similar pattern (Hills et al., 2012; Hills et al., 2015). For example, when coming up with as many animal names as possible, people may first list a number of animals that are often considered pets, then move to a patch of farm or zoo animals and then, for example, focus on various forms of felines. Hills et al. (2012, 2015) provide evidence that a patch switch in human cognitive memory retrieval occurs when patches of information or memory representations are no longer semantically similar. Semantic similarity stands for the similarity of meaning between two words or pieces of information (Hahn & Heit, 2015; Hills et al., 2012). Thus, for example, the words “cat” and “lion” are semantically similar, for they are both felines. However, the words “cat” and “crocodile” are not semantically similar, for a cat is a feline and a crocodile is a reptile. Thus, “cat” and “crocodile” do not occur in the same category (for this task). A patch switch would said to have occurred if two words are dissimilar (Troyer et al., 1997). Hills et al. 2012, tested two cognitive hypotheses on the verbal fluency task. The first one, called the static patch model, is based on the notion that a person chooses a subcategory (e.g., pets) and depletes this category before making a switch. The second method, called the fluid patch model, is based on the notion that a person switches between categories (e.g., pets and farm animals) by searching relative to the most recent term, based on similarity. The results of Hills et al. 2012 supported the fluid patch model.

Both theories of dual pathway and optimal foraging support the notion of fluid switches between cognitive categories. However, to test these theories there must be a strong operationalization on how to define cognitive categories effectively based on semantic similarity and how to classify data to multiple categories instead of one fixed category. In previous studies, this has either been done by an experimental study or, by use of outdated taxonomy or manual categorization (Dreu et al., 2011; Hills et al., 2012). However, in the current research field, there are state of the art natural language processing methods available to determine semantic similarity of words and texts, which in turn can be used to automatically categorize data based on semantic similarity. Concluding, substantial improvements could be provided to effectively categorize ideas to better understand how people come up with creative ideas.

**Natural Language Processing**

Natural language processing (NLP) is a scientific subfield of linguistics and artificial intelligence. Its main goal is to program computers to process and analyse large amounts of text data as accurately and efficiently as possible with a similar understanding of language as humans have (Manning et al., 1999). Three widely used NLP methods are “WordNet”, “Word2Vec” and “Topic Modelling”.

WordNet is a large lexical database that models the lexical knowledge of a speaker into a taxonomic hierarchy (Miller, 1995; Fellbaum, 1998). Words are organized into “synsets”, which are unordered collections of cognitive synonymous words and phrases (Handler, 2014). These synsets are in turn organized into senses, which are different meanings of the same term or concept (Varelas et al., 2005). There are a couple of relationships that WordNet can identify which are meaningful for semantic similarity. These are synonyms, hypernyms, hyponyms, meronyms and homonyms. A synonym for example is when two words have the same meaning (e.g., page and sheet) and a hyponym is when one word is more specific than the other (e.g., England and country). Thus, WordNet groups words together based on the previously named specific senses and therefore labels semantic relations among words. This could in turn be used to categorize AUT responses based on semantic similarity.

Word2Vec is a more recent unsupervised system for determining the semantic distance between words or documents. Word2Vec produces word embeddings, which are vector representations of words. These vector representations are based on the context of words. Subsequently, you can determine how close two words are to each other by looking at what other words appear in the same context. Word2Vec does not label particular semantic relationships between words (e.g., the synonym between page and sheet). Instead it assigns a number between 0 and 1, which indicates the semantic similarity between two words. Word embeddings are computed using neural networks (Zeng et al., 2014; Serban et al., 2016).

Latent Semantic Analyses (LSA) and Latent Dirichlet Allocation (LDA) topic modelling are both methods that assume that words that are close in meaning will occur in similar pieces of text. LSA uses a similarity technique which is based on word counts within a document (Gomaa & Fahmy, 2013). LDA is a method which will produce a set of clusters where each cluster represents a category containing semantically similar words (Poria et al., 2016).

In this study we chose to use Word2Vec to determine which AUT responses were most similar and belonged in the same category. This decision was based on previous studies, which have shown that representations of words learned by neural networks have high performance on similarity measurements (Mikolov, Yih & Zweig, 2013a; Peters et al., 2018). Also, Word2Vec had some advantages over other methods. For example, it performs significantly better than LSA on syntactic regularities and is computationally less expensive than LSA and LDA topic modelling (Hecking & Leydesdorff, 2018; Mikolov, Yih & Zweig, 2013a). Lastly, Word2Vec was capable of determining semantic similarities between phrases, WordNet was not capable of doing so. These considerations resulted in a final decision on Word2Vec.

**Supervised Machine Learning**

In this study, we needed a multi-label classification algorithm for we expected fluid category switches. A modern solution for multi-label and multi-class classification is neural network modeling.

A neural network (NN) is a computer system which mimics the human brain in that it “learns” to perform tasks by considering examples. This learning process is generally not programmed with task-specific rules. For example, NN is able to learn to classify images of lions and monkeys to correct classes by analyzing example images which were previously classified as lions and monkeys by human coders. The NN uses these results to identify lions or monkeys in other images. NN is considered to be a powerful method for multi-label text classification tasks (Zhang & Zhou, 2006) and therefore the logical choice for the classification problem in this study.

**Current Study**

With this study, we aim to improve the current research field of creativity. First of all, with an automated multi-label classification method, it is possible to test creative ideation theories on a widely used divergent thinking task; the AUT. Second, it will be easier to test these theories on large data, for it is a far less time-consuming method than manual categorization. Moreover, human coders are more likely to make mistakes than a computer, for humans get tired and sloppy over time (Lake et al., 2015). This will be tested by comparing between-expert reliability with prediction accuracy. In this study, we compare our automated categorization system with human experts. Several steps needed to be taken to create the automated neural network and discuss the question at hand.

The first step was to extract features, also known as predictors. This was done by use of NLP method Word2Vec. Word embeddings were extracted and were transformed to phrase embeddings. These phrase embeddings were used to calculate semantic similarities between the AUT responses. The second step was to create categories and their corresponding labels based on hierarchical cluster analysis. Clusters were determent based on semantic similarities extracted from the phrase embeddings and were labeled by an expert. The third step was to create a training dataset of categorized responses, this was done by six experts. This data was used to train a NN which was tuned on different parameters to achieve a high prediction accuracy. The final step was to test whether some of the variance between experts was taken away by the NN.

**Method**

**Sample Characteristics**

For this study, we used previously collected data in the form of the A-AUT database from the Modeling Creativity Lab at the University of Amsterdam. The A-AUT database contains a collection of AUT data obtained by different researchers in the Netherlands (Stevenson, Baas & van der Maas, 2016). This database includes over 70,000 responses to the AUT. X% of the data is from first year psychology students and Y% of the data is from high school students and Z% is from older participants.

**Materials**

***Alternative Uses Test (AUT)*** The AUT requires participants to list non-obvious uses for a common object (Guilford, 1967). In the A-AUT database participants were asked to list as many creative uses as they could for one or more of 12 everyday objects (i.e., brick, fork, newspaper, …). Participants were given a time limit of three minutes. The tests were administered on a computer, so the participants typed in each creative use on a separate line, for example they type in hammer on the first line, nutcracker on the next line, and so on. For this study we only used the data containing AUT responses from the object “brick”.

**Data Analysis and Modeling Plan**

Several steps needed to be taken to provide an algorithm which automated the categorization of the A-AUT brick database. First and foremost, data needed to be prepared and categorized by experts. Hereafter, features needed to be extracted, which were semantic similarity measurements based on word embeddings. Based on these features, clusters analysis was performed, and a neural network was trained. Lastly, the NN was compared with between-expert reliability. These steps will be described in detail in the following paragraphs.

***Data Preparation and Categorization by Experts***

The A-AUT brick database contained 23,625 responses. Of this database, we used a subset of 6,000 responses for model training. A subset of 6,000 was chosen, for this was large enough to have a representable dataset for model training. Moreover, it was a manageable size for experts to categorize. Six experts were hired to categorize the responses according to a categorization manual (Appendix A). Data was divided into two subsets of 3,000 responses, which was each categorized by three experts. Experts were not allowed to discuss category decisions with each other, so categorization was done blindly from the other experts. Also, experts were allowed to label responses to multiple categories. To combine the separately categorized responses, the modus of the three experts per dataset was calculated. The modus is the number with the highest occurrence in a vector. Every expert categorized responses to one or multiple labels. The first column of the data frame contained categories of which the experts thought they belonged best to the responses, the second column contained the second best, and so on. The modus was calculated for every column over three experts per subset of 3,000 responses. Resulting in a final categorized dataset of 6,000 responses. Analysis was conducted in R (R Core Team, 2014).

***Feature Extraction***

Features can be interpreted as predictors. In text data, for example, important predictors could be semantic information and phonetic information. Therefore, one would like to extract this information from the data and use this as input to a learning algorithm. In this study, features were derived from a large Wikipedia corpus, containing Dutch word embeddings (Bojanowski et al., 2017). The word embeddings, which were vectors of length 300, were obtained using the Word2Vec skip-gram model. We will first briefly review the skip-gram model.

Given a sequence of training words , the goal is to learn a vectoral representation for each word . Given a large training corpus represented as a sequence of words, the objective of the skip-gram model is to maximize the average log-likelihood.

(1) \_

Where c is the size of the training context (this can be a function of the centre word ). A larger c results in more training examples and leads to higher accuracy.

The essence of the skim-gram model is that it uses the context of words. For example, what words tend to appear together with the word “horse” when you look at its appearances in a large text corpus? In other words, you can guess how close two words are to each other by looking at what other words appear in the same context. This is called the distributional hypothesis.

The skip-gram model was already trained on a Wikipedia corpus; therefore, we could match the embeddings to the corresponding unique words from the AUT database. Still, we needed to transform the unique words back to the original AUT responses, which were phrases. So, we combined the word embeddings of separate words in each response using unweighted averaging. Unweighted averaging was found to do well in representations of short phrases (Mikolov, Yih & Zweig, 2013a).

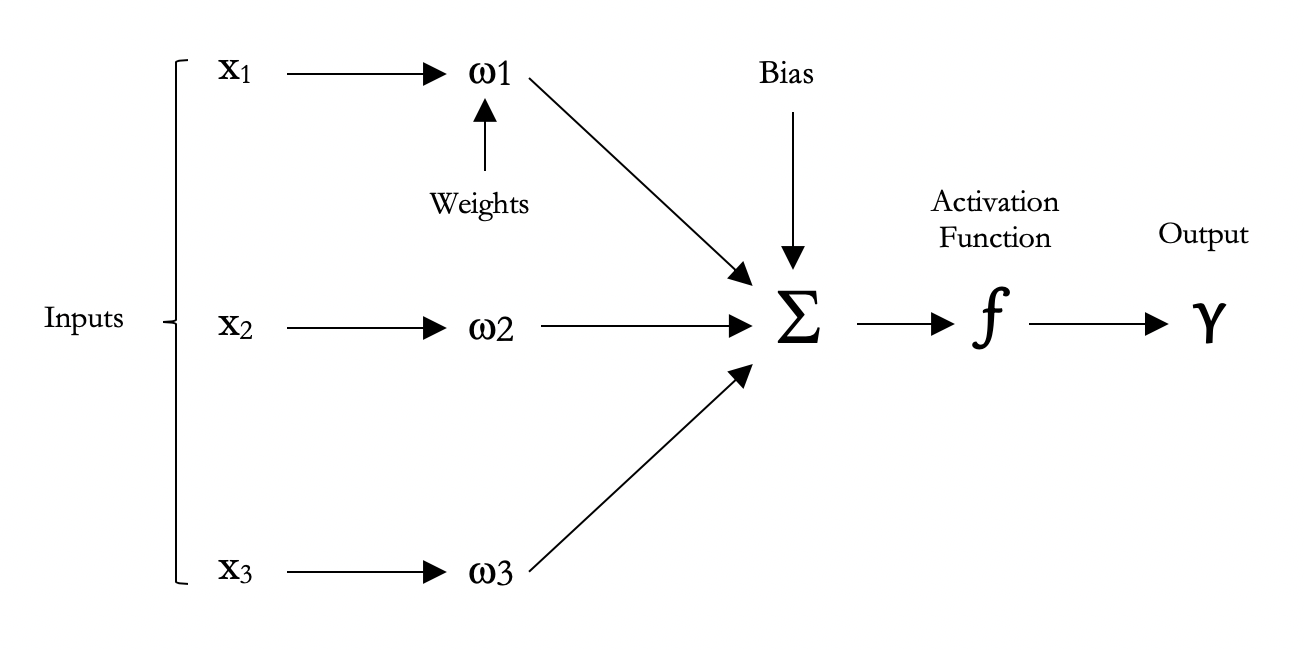
***Clustering and Labelling AUT Responses***

To decide on the right number of categories for the “brick” data, we used a hierarchical clustering algorithm. Before doing so, we already had 35 predefined categories (Baas et al., 2019). By use of a clustering algorithm we were able to see whether these categories made sense and if maybe more categories were needed. To begin with, we had much more “brick” data than the 6,000 responses we used for model training. We had a total of 23,625 uncategorized responses, which were all usable for clustering. Hierarchical clustering was performed based on semantic similarities by Word2Vec. Semantic similarities were obtained for allresponses in the same manner as previously described. The similarity matrix was transformed into a dissimilarity matrix, which could be used as input of the cluster analyses. We used bottom-up hierarchical clustering, for with this method you do not need to predefine the number of clusters. Instead, bottom-up algorithms treat each data point as a single cluster at the beginning and then merges pairs of clusters until all clusters have been merged into a single cluster containing all data points. Average silhouette method was used to determine a cut-off for the best number of clusters. This method computes the average silhouette of observations for different cluster sizes. The optimal cluster size is the one that maximizes the average silhouette over a range of possible cluster sizes. The final step was to look over all clusters and identify corresponding category labels. These labels were in turn described and used in the categorization manual for the experts.

***Neural Network Approach to Automate Categorization***

Neural network (NN) algorithms are inspired by the biological neural networks of the human brain. It is a computational system which learns to perform tasks by considering examples. NN has entered into a lot of applications; some of which are classification, pattern completion, optimization and feature detection (Dreiseitl & Ohno-Machado, 2002; Beale et al., 1996; Cochocki & Unbehauen, 1993).

A NN is constructed from three types of layers. The first one being the input layer (x1, x2 ...), this is the initial data for the neural network. The second one being the hidden layer(s), this is the intermediate layer where all computation is done. Lastly, there is the output (y) layer, this produces the results for given inputs. Each input node is connected with each node from the next layer and has a weighted association (), Figure 1. Weight can be interpreted as the impact that that node has on the node from the next layer. The bias is summed with the weighted inputs to form net inputs. The output node can range from -inf to +inf, therefore a mapping mechanism is needed between the input and the output. This mapping mechanism is known as the activation function. The output will then be transformed to an activation with a value between 0 and 1 per output node.

****

*Figure 1.* Example of a neural network with three input nodes and one output node.

Both the weights and the bias are adjustable parameters of the NN. We may describe Figure 1 and the description above in the following equation. In which the stands for the bias.

(2) \_

Next to equation one there also is an equation for the activation function, which defines if a given output node should be “activated” or not based on the weighted sum. The sigmoid function is the most common form of activation functions used in the construction of artificial neural networks (Karlik & Olgac, 2011) and represented as follows:

(3) \_

In equation 3, stands for the activation function called the sigmoid. The sigmoid function is a non-linear function with values ranging between 0 and 1.

In the current study we used a one-layer NN. The word embedding vectors were of length 300, which were given as input nodes. The output nodes were the 64 categories. We used the “nnet” package from R to train the NN. The model was fitted on a training dataset, which was 80% of all the data (N = 1392). The model was tuned on three different parameters.

The first parameter was the “size” parameter. Size stands for the number of nodes in the hidden layer. Hidden layer node size should lay between the number of input nodes and the number of output nodes (64 - 300). The more hidden nodes there are, the more computationally heavy the model becomes. Therefore, we tuned on sizes 80, 100 and 120.

The second parameter the model was tuned on was the “weight decay” parameter. This parameter needs to be tuned to reduce the risk of overfitting. The weight decay parameter is also known as L2-regularization and is represented as follows:

(4) \_

Where is a tuning parameter. Ridge regression seeks coefficient estimates that fit the data well, by minimizing RSS. The second term , called the shrinkage penalty is small when are close to zero. The tuning parameter serves to control the relative impact of those two terms on the regression coefficient estimates. When , the penalty term has no effect, and ridge regression will produce the least squares estimates. However, as , the impact of the shrinkage penalty grows, and the ridge regression coefficient will approach zero. Thus as increases, the flexibility of the ridge regression fit decreases, leading to decreased variance but increased bias. Consequently, this was an important parameter to tune. In the first round of parameter tuning we tuned on weight decays; 0.2, 0.4, 0.8 and 1. This was repeated within a smaller scope in the second round.

The third and final parameter to be tuned was the threshold, in which the threshold determines whether or not the activation value should be transformed to a 0 or a 1. A large range of values between 0.001 and .5 was tested to tune for the best threshold value.

After all these parameters were tuned on the training data and tested on the test data (20% of the data N = 350), we decided on the best model size, decay and threshold based on the prediction accuracy.

***Prediction Accuracies of NN Categorization***

To calculate prediction accuracies, we compared two binary matrixes on their agreement, these were the predicted NN category matrix and the “true” expert category matrix. For the first measure of prediction accuracy, the NN and expert matrix are multiplied resulting in a matrix containing 1’s when both matrixes contain 1’s at the same categories and 0’s when they are not in agreement or both 0. By taking the row sums of this new matrix and dividing by the row sums of the experts’ matrix we get a measure of agreement per row. By taking the mean of this row sum division, we get a matrix of agreement, which was interpreted as our first measure of prediction accuracy, see Equation 5.

(5) \_

For the second measure of prediction accuracy, the NN and expert matrix are compared to where they are equal to each other. By taking the mean of this proportion equal versus not equal, we get our second measure of agreement between the two matrices, see Equation 6.

(6) \_

In both equations *p* stands for the matrix predicted by the NN and *t* stands for the “true” expert matrix. We took both calculations of prediction accuracy into consideration when determining the best model.

***Experts versus NN Categorization***

To compare our automated NN categorization system with human experts, we tested whether the NN would take away some of the variance between expert on categorized AUT data. Measure of agreement between several experts is also called the between-expert reliability. Each experts’ categorized data was compared on their agreement with one another. Also, each experts’ categorized data was compared on their agreement with the NN. A paired t-test was computed to test whether the average agreement between experts was lower than the average agreement between experts and the NN. If the agreement between experts and the NN was higher, this would indicate that the NN explains some of the variance between experts.