Automated Categorization of the Alternative Uses Task:
A Neural Network Approach

Emma E. Schreurs

University of Amsterdam

Abstract

Previous work suggests that people come up with creative ideas by searching through a broad number of cognitive categories. Creative ideation is thought to be fluent, in that a person continuously switches between cognitive categories. Divergent thinking tasks, such as the Alternative Uses Task (AUT), are often used to study creative ideation. To test the creative ideation process categorized data is needed which is not fixed to one single category but can belong to multiple categories. With this study, we created an algorithm which automates the categorization of AUT responses. Categories were extracted by a cluster analysis of semantic similarity measurements. Semantic similarity between responses was based on Word2Vec, a natural language processing technique. Several experts categorized 6,000 AUT responses. A subset of (80%) of the data was in turn used to train a neural network. The neural network predictions for the remaining 20% of the data was compared to expert chose categories. The neural network achieved a prediction accuracy of 76%. However, the model did not remove variance between experts. With this automated multi-label classification model, we provide a pilot for a modern categorization method to more efficiently examine the creative ideation process.

Keywords: Creative Ideation, Optimal Foraging, Dual Pathway, Neural Network, Word2Vec

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

The previous paragraphs discussed NLP methods for defining semantic similarity between words, these are all unsupervised learning methods where we do not know which AUT responses belong to which categories. However, a supervised learning method, i.e., one in which we know which AUT responses belong in which categories, is needed to 'teach' a model how to categorize text data based on semantic similarity. The most common supervised classification methods are

KNN, Logistic Regression, LDA, Decision trees, Support Vector Machines and Neural Networks and we weigh their pros and cons in the following paragraphs.

K-nearest neighbour (KNN) classification is a widely used probabilistic proximity-based classifier which uses distance-based measures to perform classification (Allahyari et al., 2017). Given a positive integer (e.g., k = 1, k = 2, k = 3…) and a test observation (e.g., AUT response, a memory, someone's length), the KNN classifier first identifies the neighbouring k points in the training data that are closest to the test observation. When k = 1, the test observation will be classified to its nearest neighbouring point. When k is larger than one, the test observation will be classified to the class with the largest probability in its k nearest neighbouring points.

Logistic regression models the probability that a response belongs to a particular category. This is done by use of independent variables, which are also called predictors. For example, diet is a good predictor of weight and clouds are a good predictor of weather type. Logistic Regression is a classification method that is most suitable for two-class classification and not for multi-class classification. There are two variations to logistic regression, which are more suitable for multi-class classification. These are Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA). Of which QDA is preferred over LDA when there are many predictors and there is a lot of training data. However, LDA tends to be a better bet than QDA if there are relatively few training observations and variance reduction is crucial (James et al., 2013).

All previously named models fall under probabilistic classification methods and thus can be compared with each other. First of all, Logistic Regression can outperform LDA when the Gaussian distribution assumption is not met. However, LDA is more suitable if you have multiple predictors. KNN is a different approach, for it is non-parametric, thus no assumptions are made about the shape of the decision boundary. Therefore, KNN will outperform both Logistic Regression and LDA when the decision boundary is highly non-linear (James et al., 2013). However, KNN does not indicate which predictors are important. QDA serves as a middle ground, where the decision boundary is more flexible than Logistic Regression and LDA and it gives information about important predictors. In the end, no model definitively outperforms the other and the best model to use depends on the training data you have (James et al., 2013).

Another group of widely used classification methods falls under decision tree models. Decision trees divide training data into smaller subdivisions based on a set of tests defined at each node or branch (Allahyari et al., 2017). Popular decision trees are; Bagging, Random Forest and Boosting. Bagging works as follows, for a given test observation, the class predicted by each tree can be recorded. A majority vote can be taken, resulting in the most commonly occurring majority class among the predictors. Random Forest is a slight improvement over Bagging, in that it decorrelates the trees. Boosting is yet another method for improving the prediction accuracy of decision trees. It is a slow learning system, which reduces the risk of overfitting (James et al., 2013).

Support Vector Machines (SVM) is a supervised learning classification algorithm that has been widely used in text classification problems. It tries to find linear separators between two classes and tries to find a hyperplane with the maximum distance between the two (Allahyari et al., 2017). SVM's can be extended to multiple classes by use of one-versus-one classification or one-versus-all classification.

Even though all previously described supervised learning methods are suitable for classification, they have one limitation. They are not able to perform multi-label and multi-class classification. Class labels are absolute in that one response cannot belong to multiple classes. However, 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 two or 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 $w_1, \dots. w_T$, the goal is to learn a vectoral representation for each word w_t. 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.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} \log p\,(w_{t+j}|w_t) \qquad (1)$$

Where c is the size of the training context (this can be a function of the centre word w_t). 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 all responses 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 (x_1, x_2 ...), 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 (ω1 ...), 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.

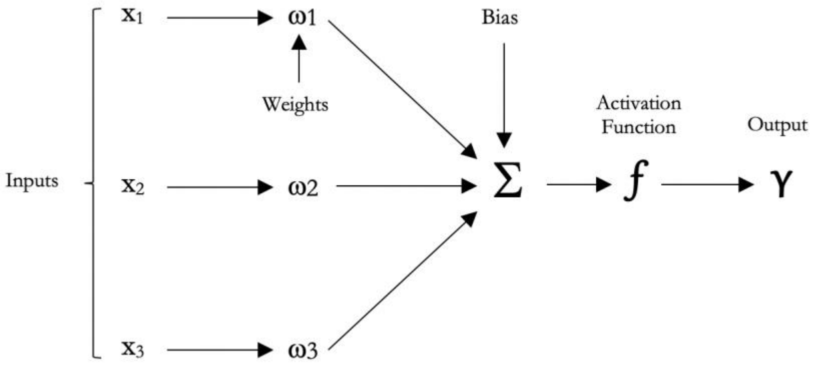


Figure 1. Example of a neural network with three input nodes and one output node.

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 θ stands for the bias.

$$\gamma_k = \sum_{i=1}^{N} (\omega_{ik} X_i) + \theta \qquad (2)$$

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)

$$h_k = \varphi(I_k)$$

In equation 3, $\varphi(I_k)$ 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:

$$\omega_j = RSS + \lambda \sum_{j=1}^{N} \beta_j^2 \qquad (4)$$

Where λ is a tuning parameter. Ridge regression seeks coefficient estimates that fit the data well, by minimizing RSS. The second term $\lambda \sum_{j=1}^{N} \beta_j^2$, called the shrinkage penalty is small when $\beta_1, \dots . \beta_j$ are close to zero. The tuning parameter λ serves to control the relative impact of those two terms on the regression coefficient estimates. When $\lambda = 0$, the penalty term has no effect, and ridge regression will produce the least squares estimates. However, as $\lambda \rightarrow \bowtie$, the impact of the shrinkage penalty grows, and the ridge regression coefficient will approach zero. Thus as λ 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.

$$\mu\left(\frac{row\sum(p * t)}{row\sum t}\right) \qquad (5)$$

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.

$$\mu\big(row\mu(p == t)\big) \qquad (6)$$

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.

Results

Clustering and Labeling

All responses to the AUT brick task, including uncategorized data, were used for cluster analysis (N = 23,625). Bottom-up hierarchical clustering with ward-linkage was performed on the cosine similarity matrix, which was extracted from Word2Vec word embeddings. Average Silhouette was performed to extract and optimal cluster size (k). There were 35 predefined category labels, therefore we looked for an optimal average silhouette width near 35 clusters. A maximum was found at 64 clusters, Figure 1.

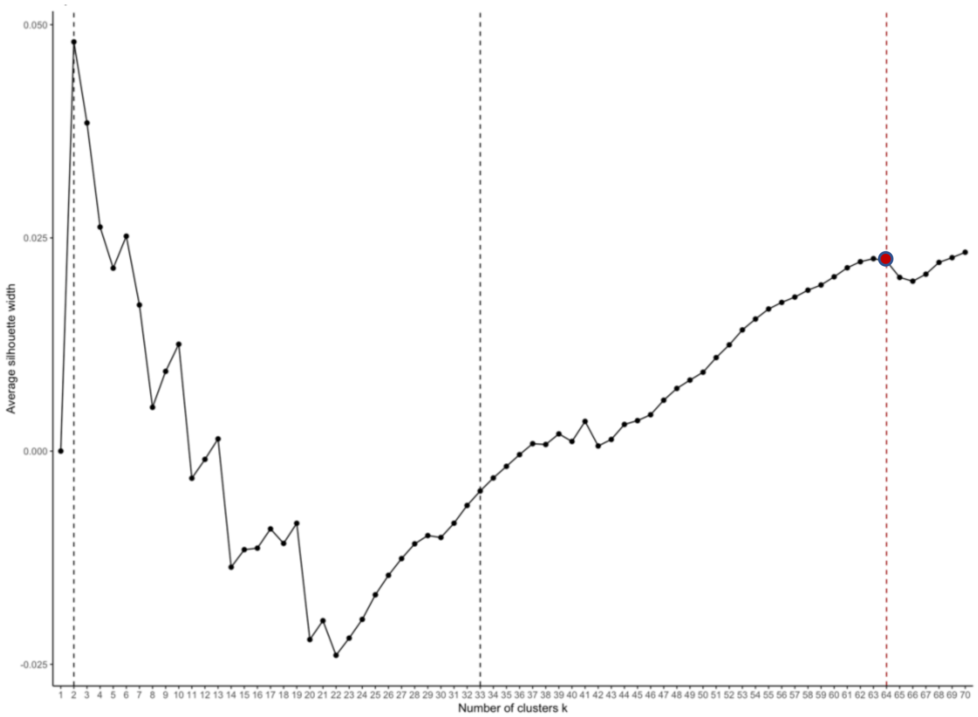


Figure 1. Average Silhouette Method visualized for determination of the optimal number of clusters (k). The first black dashed line represents the highest average silhouette width, the second dashed line the already excising 35 clusters and the red dashed line the number of clusters chosen for this study.

The corresponding responses were extracted for each cluster and the predefined 35 category labels were matched to 35 of the clusters we found. Then new category labels were created for the remaining 29 clusters. Table 1 contains one of the 64 clusters with a subset of its corresponding responses. These responses belong to category label "Animal accessories"

Table 1

The first four responses belonging to cluster 41 from the cluster analysis. These were assigned to category label "Animal accessories"

Response	Cluster Category
Rabbit cage	41
Brick cage pet	41
Cage	41
Cabin pet	41

Descriptives

Data was categorized into 64 categories by 6 experts, where each response could belong to more than one category. Figure 2 demonstrates the frequency with which responses were categorized by experts to each of the 64 categories. As you can see, some categories were hardly used while others were high in demand. For example, category one (building) was a frequently used category. Next to this, category zero (other) was also frequently used, i.e. 179 responses (10 %) did not belong to any of the 63 labelled categories. More importantly, there were quite a few categories that were hardly ever used (e.g., Technology, Cutting, Filling and Science). One category was never used, this was category 32 (Teasing).

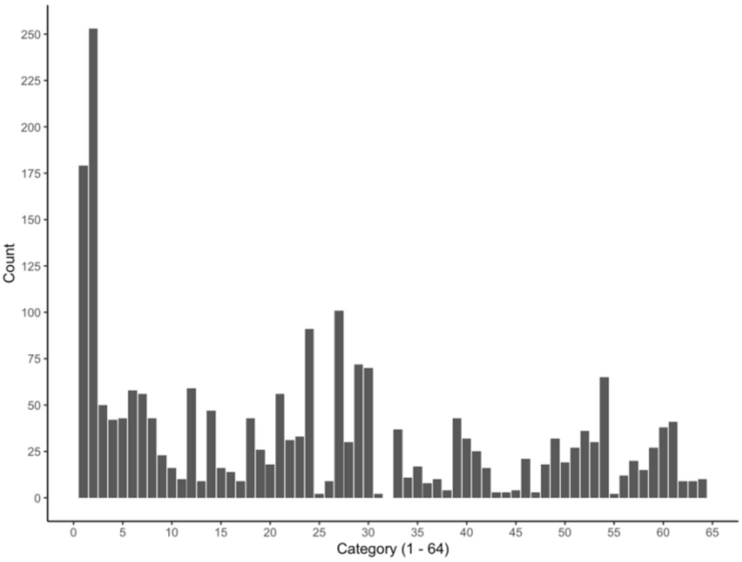


Figure 2. Frequencies of categories used over all 1,741 responses.

Neural Network Model Selection

After data cleaning and selecting only unique responses, a categorized dataset of N = 1,741 remained, where N = 1,392 (80 %) was used to train the neural network. Size, decay and threshold parameters were tuned for model selection, where three size values (80, 100 and 120) and five decay values (0.2, 0.4, 0.6, 0.8, 1) were applied. This resulted in 15 models. Each of these 15 models were cross-validated on the remaining 20% of the data, which was a subset of 350 responses. Cross-validation was performed on 250 threshold values for each of the 15 models, resulting in 3,750 models. For each model, two measures of accuracy were calculated, the sum accuracy and the mean accuracy (see section "*Experts versus NN Categorization*"). These measurements were combined into an unweighted average, referred to as prediction accuracy. Table 1 demonstrates the top ten models based on predictions from all 3.750 cross-validated models. The neural network model with a size of 100, a decay of 0.2 and threshold of 0.031 performed best.

Table 1

Top 10 performing neural networks, tuned of size, decay and threshold and cross-validated on the sum accuracy, the mean accuracy and the prediction accuracy (which is the unweighted average of the sum and mean accuracy). The Table is arranged from highest prediction accuracy to lowest and is a subset of the 10 first rows of a table of 3750 rows.

Model	Size	Decay	Threshold	Sum Accuracy	Mean Accuracy	Prediction Accuracy
1	100	0.2	0.031	0.686	0.837	0.761
2	100	0.2	0.029	0.699	0.824	0.761
3	100	0.2	0.041	0.634	0.882	0.758
4	100	0.2	0.042	0.626	0.888	0.757
5	80	0.2	0.023	0.731	0.782	0.757
6	100	0.2	0.039	0.637	0.876	0.756
7	80	0.2	0.021	0.759	0.754	0.756
8	100	0.2	0.033	0.663	0.849	0.756
9	80	0.2	0.025	0.706	0.806	0.756
10	100	0.2	0.045	0.617	0.894	0.756

The reasoning behind the choice of the best model can be seen in Figure 3. For the Size parameter sizes 80 and 100 did not differ much in performance (Figure 3, A). Size 80 though has more variability than 100, whereas size 100 has more outliers. For the Decay parameter the value of 0.2 has the highest median prediction accuracy (Figure 3, B). The higher the decay becomes, the lower the prediction accuracy is. But, for decay is 1, the prediction accuracy increases somewhat.

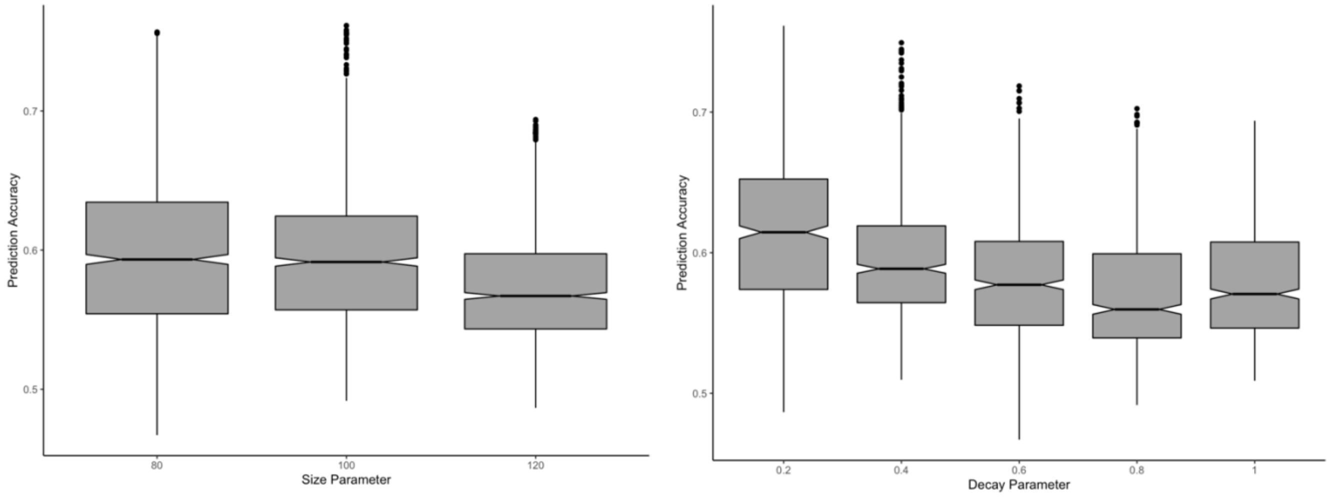


Figure 3. Boxplots for size and decay parameter tuning. Plot A, size paramater 80, 100 and 120. Plot B, decay parameters ranging from 0.2 to 1.

Another parameter of importance for the choice of the best model was the threshold parameter, which can be seen in Figure 4. For each model, the prediction accuracy peaks around 0.031 (red dashed line) and steadily decreases after (Figure 4). Threshold 0.031 resulted in the best performing model (see red dot, Figure 4).

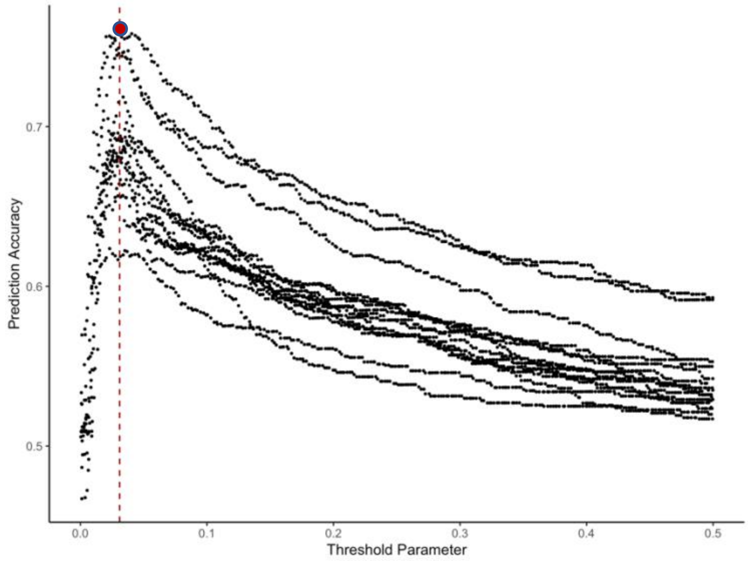


Figure 4. 250 threshold values ranging between 0.001 and 0.5 for all 15 models and the corresponding prediction accuracies. Red dashed line and red dot represents the optimal threshold value (0.031) for the best neural network.

Five more models were trained on size 100 and decay values (0.05, 0.10, 0.15, 0.20, 0.25). All models were cross validated on the same test data for all 250 thresholds values. A decay of 0.20 resulted in the highest prediction accuracy. To test whether a lower size than 100 and 80 would perform better, two more models were trained on sizes 70 and 90 and decay of 0.20. The final best performing model was selected; size = 100, decay = 0.20 and threshold = 0.031.

Experts versus NN Categorization

To gain insight into how well the NN categorized the AUT responses expert reliability was compared to the NN model versus expert reliability. Between expert reliability was calculated by comparing the binary matrixes of the two experts. The mean of the vectors was taken, resulting in an average between expert reliability for the entire dataset. Model versus expert reliability was calculated by comparing the model predictions with each of the two experts. The mean was taken of these three vectors, resulting in an average model versus expert reliability. A paired t-test was computed for the two resulting vectors. The mean of the between expert reliability was significantly higher (M=63) than the expert versus model reliability (M=53; t = -62.3, p < 0.001), as can be seen in Figure 5. Thus, the NN did not take away any of the variance between experts.

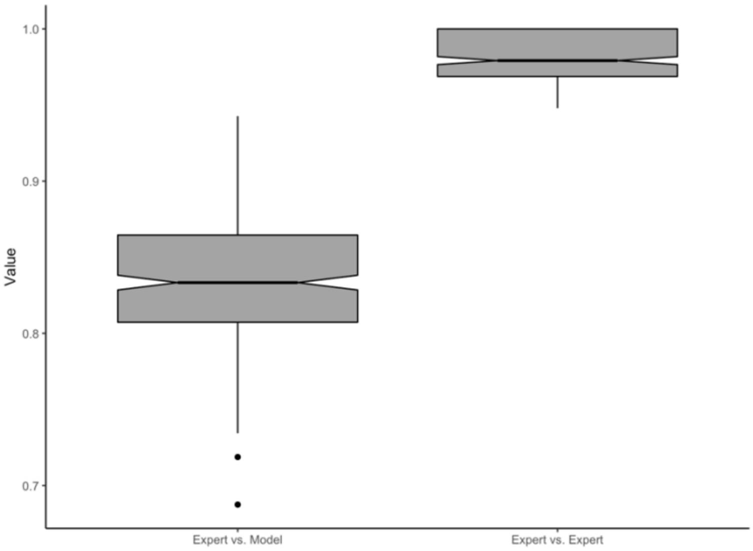


Figure 5. Expert versus model compared to expert versus expert reliability.

Discussion

With this study, we provided an automated multi-label classification algorithm for AUT responses. A neural network approach was used to predict categories based on phrase embeddings, which provided an indication of how semantically similar each pair of AUT responses were. Our automated multi-label classification algorithm achieved a prediction accuracy of 76 percent, so approximately three quarters of the test data was predicted correctly by our model. This result provides a stepping stone in the use of modern computer science automation for studies in the field of creativity. Our results are suitable to test cognitive flexibility and optimal foraging theories more effectively, which are both hot topics in the field of creativity research (Hass, 2017; Hills et al., 2012; de Dreu et al., 2011). Previous studies used hand-coded categories and subcategories and extracted semantic similarities between responses from the BEAGLE model (Hills et al., 2012; Troyer et al., 1997). With this study, we proposed a solution to hand-coding large datasets and provided a modern method for similarity measurements (Word2Vec). Moreover, we were able to create an algorithm which could deal with a multi-label problem, which is important when testing for fluid category switches (Hills et al., 2012).

Regarding model versus human performance, between expert reliability was not improved by the neural network, meaning it did not take away any of the variance between experts. Between expert reliability was 63% versus between model and expert reliability which was 53%. This could be explained by the fact that the neural network is trained on an averaged training set from all the experts, resulting in more variation when compared to each individual expert. Nevertheless, machine learning algorithms are known to outperform humans', especially when considering large repetitive tasks of which humans' get tired more easily (Lake et al., 2015). In the current study, we only looked whether the neural network model takes away some of the variance between experts. This is not a perfect indication of whether the model outperforms humans' or not. An experimental study could be done, in which the percentage error from the model is compared to the percentage error of a human (e.g., Lake et al., 2015). However, to do so, you would need a "perfect" training and test dataset. This requires extensive reviews of the training and test data by experts with use of a strong operationalization. This is also of great importance for achieving high prediction accuracies (Langley, 1994).

Regarding prediction accuracies, there are some improvements to be made. In this study, we were able to create a neural network classification algorithm which performed reasonably well. Three quarter of the test data was classified into the right categories. However, to generalize this algorithm to classify all 23,625 AUT brick responses we would like to achieve an accuracy that is as high as possible.

Limitations and Future Directions

There were some limitations to this study, which by improvement would most likely improve the model's predictions. First, the hierarchical clustering algorithm had some limitations in its input, the interpretation of the output and the labeling of the clusters. The inputs to the clustering algorithm were phrase embeddings, which were calculated based on an unweighted average of the word embeddings. However, when eyeballing the clusters, it was clear that responses in clusters were often based on one word being very similar. For example, the word "paper" would occur very often in cluster 47. Yet, for some responses the word "paper" was used in that the brick would prevent the paper form blowing away and in other responses the word "paper" was used as wrapping paper for the brick. This implies that an unweighted average for phrase embeddings does not result in the best similarity measurements. It could be that verbs should have more weight than nouns. A weighted average using TF-IDF weights could improve similarity measurements between phrase embeddings and could therefore improve cluster analysis (Elsaadawy et al., 2018; Arora at al., 2016; Mikolov et al., 2013). Also, the decision on cluster size remains disputable. In this study, average silhouette was used to choose an optimal cluster size. This method computes the average silhouette of observations for different values of k (number of clusters). The optimal number of k is the one that maximizes the average silhouette over a range of possible values for k. Average silhouette did not provide one clear peak around size k that we expected, either very high or very low. However, we needed to take into account that the cluster size could not be too large, for it would become very time consuming for the experts to classify the data. The more clusters there are the harder it is for people to correctly assign responses without making mistakes or forgetting certain categories. A decision on the right size of k could also be improved by using other methods to determine the optimal size of k (e.g., Elbow method, Gap statistic and Sum of Squares method). By looking at all of the results from these methods and comparing them with each other, a consensus could be found on a size k. This would be more reliable than only using average silhouette. Another set for determining the number of clusters are information criteria, The Akaike information criterion (AIC), The Bayesian information criterion (BIC), The Deviance information criterion (DIC) (Madhulatha, 2012). Lastly, the labeling of the clusters could be improved by letting multiple experts label the clusters and find a consensus on each cluster label.

A second limitation of this study is that we took the ranking order of the categories into account. If a response belonged to multiple categories, the first category should represent the best fitting category. However, by taking the modus between experts, we lost some of the categories where there was no consensus between experts. Even though these categories might have fitted the responses. This could be improved by not taking into account the order of the categories, but instead including all categories the experts classified. This could in turn improve prediction accuracies.

A third limitation is our choice of semantic similarity measurement, Word2Vec, which is a modern way of extracting word embeddings from text data. Word2Vec was pre-trained on a large Dutch Wikipedia corpus. To improve prediction accuracy, it is important to extract good features (Langley, 1994). In this case, feature extraction could be improved by improving the word embeddings and thus phrase embeddings. This could be done by, for example, using a better text corpus. It would be good to use a corpus in which words are used in different contexts (e.g., books, subtitles, reviews). A combination of different corpuses is found to perform less than a single suitable corpus. Moreover, the size of the corpus plays a large role in its performance (Tulkens, Emmery & Daelemans, 2016). Another way of obtaining optimal phrase embeddings would be to use different methods of extracting word embeddings (e.g., Word2Vec, GloVe, BERT, ELMo). And compare model performance based on these different NLP methods.

Conclusion

This study has several key contributions. We demonstrated how to use Natural Language Processing as an important tool in semantic similarity measurements, which can be applied to studies related to creativity (e.g., Optimal Foraging, Dual Pathway and other Creative Ideation processes). In addition, we propose a data driven decision (cluster analyses) on the number of categories within a large database, based on semantic similarity measurements. Moreover, a Neural Network classification algorithm was created, which automates the classification of AUT responses. This algorithm takes away many hours spent on manual classification and provides a quick and accurate solution for testing theories on large datasets. More specifically, we created an algorithm which provides a large dataset of categorized AUT responses. Future studies could investigate both the dual pathway to creativity model (de Dreu et al., 2008; de Dreu et al., 2011) and the optimal foraging theory (Hass, 2017; Hills et al., 2012; Fu & Pirolli, 2007; Hills et al., 2015; Payne, Duggan, & Neth, 2007) using this data and would therefore provide a fine-grained contribution to our understanding of the divergent thinking process.

References

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*.

Beale, H. D., Demuth, H. B., & Hagan, M. T. (1996). Neural network design. *Pws, Boston*.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135-146.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., & Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems* (pp. 288-296).

Charnov, E. L. (1976). Optimal foraging, the marginal value theorem.

Cochocki, A., & Unbehauen, R. (1993). *Neural networks for optimization and signal processing*. John Wiley & Sons, Inc.

Dippo, C., & Kudrowitz, B. (2013). Evaluating the alternative uses test of creativity. *2013 NCUR*.

Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of biomedical informatics*, *35*(5-6), 352-359.

De Dreu, C. K. D., Nijstad, B. A., & Baas, M. (2011). Behavioral activation links to creativity because of increased cognitive flexibility. *Social Psychological and Personality Science*, *2*(1), 72-80.

De Dreu, C. K., Baas, M., & Nijstad, B. A. (2008). Hedonic tone and activation level in the mood-creativity link: toward a dual pathway to creativity model. *Journal of personality and social psychology*, *94*(5), 739.

Elsaadawy, A., Torki, M., & Ei-Makky, N. (2018, December). A Text Classifier Using Weighted Average Word Embedding. In *2018 International Japan-Africa Conference on Electronics, Communications and Computations (JAC-ECC)* (pp. 151-154). IEEE.

Fu, W. T., & Pirolli, P. (2007). SNIF-ACT: A cognitive model of user navigation on the World Wide Web. *Human–Computer Interaction*, *22*(4), 355-412.

Guilford, J. P. (1967). Creativity: Yesterday, today and tomorrow. *The Journal of Creative Behavior*, *1*(1), 3-14.

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. (2013). *An introduction to statistical learning: with applications in R*. New York: Springer, 2013

George A. Miller (1995). WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41.

Gomaa, W. H., & Fahmy, A. A. (2013). A survey of text similarity approaches. *International Journal* *of Computer Applications*, *68*(13), 13-18.

Hass, R. W. (2017). Semantic search during divergent thinking. *Cognition*, *166*, 344-357.

Hecking, T., & Leydesdorff, L. (2018). Topic Modelling of Empirical Text Corpora: Validity, Reliability, and Reproducibility in Comparison to Semantic Maps. *arXiv preprint arXiv:1806.01045*.

Hills, T. T., Jones, M. N., & Todd, P. M. (2012). Optimal foraging in semantic memory. *Psychological review*, *119*(2), 431.

Hills, T. T., Todd, P. M., & Jones, M. N. (2015). Foraging in semantic fields: How we search through memory. *Topics in Cognitive Science*, *7*(3), 513-534.

Karlik, B., & Olgac, A. V. (2011). Performance analysis of various activation functions in generalized MLP architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, *1*(4), 111-122.

Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, *350*(6266), 1332-1338.

Langley, P. (1994, November). Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall symposium on relevance* (Vol. 184, pp. 245-271).

Madhulatha, T. S. (2012). An overview on clustering methods. *arXiv preprint arXiv:1205.1117*.

Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

Mikolov, T., Yih, W. T., & Zweig, G. (2013, June). Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 746-751).

Payne, S. J., & Duggan, G. B. (2011). Giving up problem solving. *Memory & cognition*, *39*(5), 902-913.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Poria, S., Chaturvedi, I., Cambria, E., & Bisio, F. (2016, July). Sentic LDA: Improving on LDA with semantic similarity for aspect-based sentiment analysis. In *2016 international joint conference on neural networks (IJCNN)* (pp. 4465-4473). IEEE.

R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/

Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., & Pineau, J. (2016, March). Building end-to-end dialogue systems using generative hierarchical neural network models. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Silvia, P. J., Winterstein, B. P., Willse, J. T., Barona, C. M., Cram, J. T., Hess, K. I., ... & Richard,

C. A. (2008). Assessing creativity with divergent thinking tasks: Exploring the reliability and validity of new subjective scoring methods. *Psychology of Aesthetics, Creativity, and the Arts*, *2*(2), 68.

Stein, M. I. (1953). Creativity and culture. *The journal of psychology*, *36*(2), 311-322.

Stevenson, C. E., Baas, M., van der Maas, H. L. J. (2016). The Amsterdam Alternative Uses Task (A- AUT). Consulted from http://modelingcreativity.org/aut-database/

Tulkens, S., Emmery, C., & Daelemans, W. (2016). Evaluating unsupervised Dutch word embeddings as a linguistic resource. *arXiv preprint arXiv:1607.00225*.

Varelas, G., Voutsakis, E., Raftopoulou, P., Petrakis, E. G., & Milios, E. E. (2005, November). Semantic similarity methods in wordNet and their application to information retrieval on the web. In *Proceedings of the 7th annual ACM international workshop on Web information and data management* (pp. 10-16). ACM.

Zhang, M. L., & Zhou, Z. H. (2006). Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering*, *18*(10), 1338-1351.

Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. (2014). Relation classification via convolutional deep neural network.

Appendix A

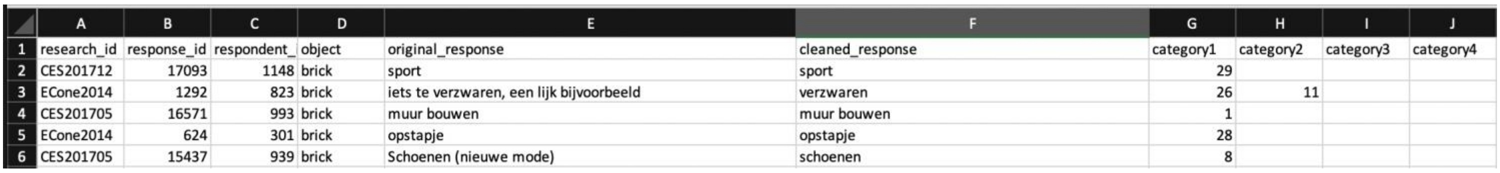
Protocol Categorizing AUT Brick Responses

For this study I will need a total of 6,000 categorized AUT responses from the brick data. This data will be split into two sets of 3,000 responses. For each dataset, there will be 3 experts which will categorize the data. These experts are not allowed to look at one another's categorization, thus it will be done blindly from each other.

<u>Step by step approach:</u>

- When you open the dropbox, you will find a file with your name (e.g., Cat_01_Subset1_YourName). In this file you find the document containing the data (data_subset.xls) which you can open in excel on your computer. Never use a file from another expert's folder.
- In the .xls file you will see that there are 10 columns, of which the final 4 columns are empty with the name "category1", "category2", etc. These columns will need to be filled with the categories for the responses.
- On the next page, you can find all possible categories for the data (63 categories). A response can belong to 1 category, but possibly to more (definitely not always the case). When this is the case you add the extra category to column "category2" or column "category3 when the response belongs to three categories. Always fill the best fitted category in at the first column: "category1". Hereafter the second best fitted, and so on.
- It is MOST important that you fill in the numbers corresponding to the categories and not the actual names of the categories.
- You will categorize 3,000 responses.
- Please safe the file as an .xls file and upload it in the dropbox.
- Thanks a lot!

A short example of what it should look like!



	A	B	C	D	E	F	G	H	I	J
1	research_id	response_id	respondent_	object	original_response	cleaned_response	category1	category2	category3	category4
2	CES201712	17093	1148	brick	sport	sport	29			
3	ECone2014	1292	823	brick	iets te verzwaren, een lijk bijvoorbeeld	verzwaren	26	11		
4	CES201705	16571	993	brick	muur bouwen	muur bouwen	1			
5	ECone2014	624	301	brick	opstapje	opstapje	28			
6	CES201705	15437	939	brick	Schoenen (nieuwe mode)	schoenen	8			

1. Bouwen
2. Gooien
3. Vandalisme
4. Agressie
5. Meubels
6. Huis accessoires
7. Tekenen/Schrijven
8. Kleding
9. Voedsel
10. Cosmetica Accessoire
11. Gereedschap
12. Technologie
13. Component/Materiaal
14. Muziekinstrument/Geluid maken
15. Voertuig
16. Item met emotionele waarde
17. Kunst
18. Verkeer
19. Game
20. Steunen
21. Blokkeren
22. Territorium markeren
23. Kapotslaan/Breken
24. Snijden
25. Opvullen
26. Als gewicht
27. Bescherming
28. Verhoging
29. Sporten
30. Therapie
31. Plagen
32. Hitte/Vuur
33. Ergens instoppen
34. Fantasie
35. Schoonmaken
36. Ruilen/geld
37. Wetenschap
38. In de keuken
39. Accessoire voor tuin
40. Bedekken/Afsluiten
41. Accessoire voor dier
42. Feestdagen
43. Verjaardagen
44. Graven
45. Moord
46. Overleven
47. Balanceren
48. Versieren
49. Beeldhouwen
50. Wapen
51. Speelgoed
52. Sociale interactie
53. Hulpmiddel/Tool
54. Sfeer
55. Pletten
56. Baan
57. Kleur
58. Vermalen
59. Knutselen
60. Vermaak
61. Natuur
62. Testen
63. Water tool
64. Overig