Proceedings of the Twenty-Ninth International
Florida Artificial Intelligence Research Society Conference

Maximizing the Number of Appropriate Responses Returned by a Conversational Agent through the Use of a Genetic Algorithm for Feature Selection

Jerome L. McClendon, Larry F. Hodges, Sekou L. Remy

Abstract

We present an approach to creating conversational agents that are capable of returning appropriate responses to natural language input. The approach described consists of a genetic algorithm used as a feature selection technique to evolve a subset of random features towards a set of features that are more relevant to the language used in the domain; therefore improving the conversational agent's ability to return appropriate responses. The results show that over multiple iterations of the evolutionary process the genetic algorithm was able to filter out unfit features. After the evolutionary process the features that were found to be relevant were tested on an unseen test set and the algorithm achieved an accuracy of 72.678%.

Introduction

Previous studies done by researchers on conversational agents found that if the agent is unable to produce a relevant response to the user's last statement, the user may become frustrated and is less likely to use the system (Bloodworth et al. 2012). To increase the number of relevant responses returned by the agent one approach used by researchers is to model the set of statements that they anticipate the user will say (Leuski et al. 2009). These anticipated statements are referred to as queries and are stored in a data structure known as the script. Once the initial set of queries has been created researchers then create paraphrases of each query to capture variation in the user's statements. A query and its corresponding paraphrases are referred to as a class.

Each class in the script is paired with a response. This response is the appropriate expression to be returned by the conversational agent when presented with the user's statement. The appropriate response is determined by using a classification model. The classification model calculates the probability of assigning the user's last statement to a class given a set of natural language processing features. A drawback for conversational agents that use this approach is that the classification model used does not evaluate the features' importance. This can lead to the use of irrelevant features during training and classification (McClendon, Mack, and Hodges 2014). Irrelevant features will affect the classification performance by creating noise that results in a decrease in accuracy or, as it relates to conversational agents, the retrieval of inappropriate responses.

To maximize the number of appropriate responses returned by the conversational agent we have implemented a genetic algorithm for selecting a subset of relevant features to be used by the classification model. In the following subsections, a description is given of the classification model and genetic algorithm used for feature selection.

Classification Model and Features

To classify the user's last statement into a class we have chosen to implement n-gram language model. In natural language processing an n-gram is comprised of *n* words that appear beside each other. The n-gram model is defined by the equation:

$$p(s) = \prod_{i=1}^{n} p(n\text{–}gram_i|c_j) \tag{1}$$

Where:

- $n\text{–}gram$ is the feature set consisting of $\{uni\text{–}grams, bi\text{–}grams, tri\text{–}grams, quad\text{–}grams\}$
- $n\text{–}gram_i \in n\text{–}gram$
- $p(n\text{–}gram_i|c_j)$ is the conditional probability of this n-gram appearing in this class

To compute the conditional probability the maximum likelihood estimation (MLE) is used. MLE is defined by the equation:

$$p(n\text{–}gram_i|c_j) = \frac{count(n\text{–}gram_i \wedge c_j)}{|c_j|} \tag{2}$$

A weakness of MLE is that common terms might be incorrectly emphasized. To solve this problem we have included in the equation a weight that counts the number of classes $n\text{–}gram_i$ appears in. We refer to the modified MLE as the weighted maximum likelihood estimation (WMLE):

$$wp(n\text{–}gram_i|c_j) = (p(n\text{–}gram_i|c_j) * \frac{1}{count(\{c_k \in C : n\text{–}gram_i \in c\})}) \tag{3}$$

Where :

- $p(n\text{–}gram_i|c_j)$ is the conditional probability of this n-gram appearing in this class
- $count(\{c_k \in C : n\text{–}gram_i \in c\})$ is the number of classes, $n\text{–}gram_i$ appears in.

The uni-grams "cry" and "tears" are semantically similar. However, they consist of different phonemes and the classifier will view these uni-grams as being different when calculating the frequency. To solve this problem, a semantic similarity technique has been incorporated that utilizes a lexical ontology to create a weighted graph of semantic relationships. This graph will be used to measure the similarity of n-gram features. Using the weighted graph the classifier will determine if it is possible to find a path between two n-grams based on a depth parameter. The depth values used are 1 and 2. When the value is 1 the classification model will use the relationships in the weighted graph to perform a depth search at level one. If a path is found, then the two n-grams are considered similar. When the value is 2 the classification model will perform a depth search at level 2.

Setting the depth parameter to 1 or 2 will reduce the number of concepts that are found to be similar. However, it is still possible that the lexical chain method will incorrectly identify concepts as being similar. To improve the accuracy of the lexical chain method, every path is assigned a value which is compared to the word-similarity threshold. The value associated with the path is referred to as the cost. If the cost of the path is greater than the word-similarity threshold, then the path is accepted and the concepts are said to be similar. The word-similarity threshold must be selected from a range of [0.24, 0.90]. This range is defined by the minimum and maximum cost for a generated path.

When the user's statement is outside of the expected domain of the conversation then the appropriate response should be equivalent to an off topic response such as, "I do not understand." To solve this problem, a similarity threshold must be set. If the probability score does not exceed this threshold then the conversational agent should return the off topic response. This similarity threshold is referred to as the acceptance threshold. The acceptance threshold value must be selected from a range of [0.00, 0.99].

The problem with the classification model described is that there is no method for determining which subset of features and thresholds will optimize the accuracy of the classification model for a given script. It is impossible to guarantee that bi-grams and a depth value of 1 for measuring similarity will result in reasonable accuracy. It is also not guaranteed that the weighted maximum likelihood estimate will produce better results than the maximum likelihood estimate. Therefore, a feature selection technique is used to select a relevant subset of features and threshold values to use when maximizing the number of appropriate responses returned by the classification model.

Feature Selection

To optimize the accuracy of the classification model a genetic algorithm has been employed in determining a subset of relevant features to be used. This process begins with generating a random population of solutions. In the literature solutions are referred to as chromosomes. Chromosomes are made up of separate individual genes. Genes represent variables that are relevant to the optimization problem being solved. As it relates to this research the set of features given to the classification model have been encoded as genes.

After the generation of the random population is completed each chromosome is evaluated using a a metric that characterizes the performance of a chromosome called a fitness function. Once the fitness of each chromosome has been evaluated, then the highest fit chromosomes are chosen for recombination (reproduction). Recombination is carried out using genetic functions that have been inspired from the theory of biological evolution. The offspring created during recombination will replace the less-fit members of the population creating a new population. The algorithm then starts over with the new population and the process continues until it reaches a termination point decided by the developer.

The fitness function as it relates to this research can be defined by the following equation:

$$Fitness = \frac{TP + TN}{TP + TN + FPI + FPII + FN} \quad (4)$$

- TP: The count of how many times the conversational agent returns an appropriate response from the script.
- TN: The count of how many times the conversational agent returns "I do not understand." and the appropriate response to that query was not present in the script.
- FN: The count of how many times the conversational agent returns "I do not understand." when an appropriate response was stored in the script.
- FPI: The count of how many times the conversational agent returns an incorrect response from the script when an appropriate response is present in the script.
- $FPII$: The count of how many times the conversational agent returns a response from the script when it should not have, because the true value was "I do not understand."

As stated earlier chromosomes consist of genes. Genes used in this research were:

- Genes 0-3 encodes the n-gram features {uni-grams, bi-grams, tri-grams, quad-grams}. The value for the genes is binary and is either $\{0, 1\}$. When the value of the gene is 0 the classification model will not include that specific n-gram. When the value of the gene is 1, that specific n-gram will be included by the classification model.
- Gene 4 encodes the acceptance scalar. The value for the acceptance scalar can be any number from the interval [0.00, 1.00]. The difficulty in assigning an acceptance threshold is in determining the correct threshold value that indicates that these two statements share enough information to be labeled as similar (Dickerson et al. 2005). Therefore, a optimal threshold value must be encoded as a gene. To find an appropriate acceptance threshold for each class requires that each class

be encoded as a gene. This would allow the genetic algorithm to learn an appropriate value for each class. The problem with this approach is that as the number of classes for the script increases the optimization time increases. This causes the accuracy of the classification model to decrease. To solve this problem a single scalar value was encoded as a gene. This gene is multiplied against a pre-defined acceptance threshold set for that class before optimization. With this approach the genetic algorithm learns the appropriate scalar value for the entire script.

- Gene 5 encodes the term-weight parameter. The value for this parameter is binary and is either $\{0, 1\}$. When the parameter is 0 the classification model will calculate the probability using the maximum likelihood estimation. If the parameter is 1 the classification model will calculate the probability using the weighted maximum likelihood estimation.
- Gene 6 encodes the depth parameter. The value for this parameter can be either $\{0, 1, 2\}$. The classification model will perform a depth search at level one when the value of the parameter is 1 and a search at level two when the depth parameter is 2. When the depth parameter is 0 the classification model will not account for how semantically similar n-grams are. This means n-grams have to be phonetically the same to say that they are equivalent.
- Gene 7 encodes the word-similarity threshold. The value for this parameter can either be a floating point number from the interval [0.24, 0.90] or the integer value of 1. The word-similarity threshold is set to 1 only if the value from depth parameter is 0. When the word-similarity threshold is 1, the n-grams have to be phonetically equivalent. In the case where the word-similarity threshold value is not 1 then the classification model will compare the semantic similarity score produced by searching the weighted graph to the word-similarity threshold. If the similarity score is greater than the threshold, the two n-grams are similar enough to be replaced with each other within a sentence.

Genetic Algorithm Evaluation

To evaluate the genetic algorithm ability to optimize the accuracy of the classification model through feature selection a script was created. The data for the script came from a previous conversational agent project. The script consisted of 172 unique query/response pairs. Of the 1,519 pairs 1,347 were used during the evolutionary process. The remaining 172 queries were used in an unseen test set along with 11 random queries. These 11 queries were taken from a set of 1,000 questions that were picked from the web and were not related to the questions contained in the script. To ensure that the queries within the classes were paraphrases of each other the quality of the script was verified by raters.

During the evolutionary process, 3-fold cross validation was used to create training/validation data sets. For each fold the training set consisted of 953 queries and the validation set consisted of 434 queries. Out of the 434 queries in the validation set, 394 of the queries were taken from the script and the remaining 40 queries were taken from the set of 1,000 random questions.

There were 60 populations generated over the entire evolutionary process. This number of generations was also used as the termination point for the optimization process. Each generation of a population consisted of 80 chromosomes.

Results and Discussion

In the feature selection stage, the genetic algorithm performs a search through the feature space proposing a subset of new features and evaluating that subset on training data taken from the script. As the genetic algorithm searches through the feature space the performance of the classification model is evaluated. This evaluation is done by determining if the subset of features given to the classifier improves in fitness over multiple generations of populations. When comparing the fitness of the chromosomes within later populations to earlier populations it has been determined that the subset of features given to the classification model improved in fitness over multiple generations. The box and whisker plot shown in Figure 1 provides evidence that the subset features given to the classification model has improved in fitness over multiple generations.

When analyzing figure 1 it can be seen that over multiple generations of the population there is an increasing trend with respect to the fitness scores of the chromosomes within the population. This trend can be seen across multiple generations as the median shifts from the middle of the box to the upper quartile and the distributions become positively skewed with a high concentration of scores in the upper part of the scale.

The evidence suggests that the increasing trend was caused by the classifier receiving an improved subset of features over multiple generations. This is supported by the implementation of the genetic algorithm. During the optimization process the genes of the highest fit chromosomes have a better chance of being included in future populations than the genes of the less fit. This is because the genes from the higher fit chromosomes are intermixed using the crossover function that produces offspring. These offspring replace the less fit chromosomes. Therefore the genetic algorithm is filtering out the irrelevant features by replacing the unfit chromosomes with chromosomes that have high fitness.

After the 60 populations were evaluated the genetic algorithm then selected the 80 chromosomes with the highest fitness scores observed during the entire evolutionary process. These 80 chromosomes make up population 61. The following list describes the genetic makeup of population 61:

- Out of the 80 chromosomes 79 of them consisted of uni-grams. The other 1 chromosome consisted of $\{uni\text{–}grams, bi\text{–}grams\}$ After analyzing the results, it was concluded that since bi-grams, tri-grams and quad-grams are made up of individual uni-gram features these features are unlikely to outperform uni-grams. The only instance where uni-grams can be outperformed by a high order feature is when there exist multiple classes that consist of the same uni-grams, but the queries within the

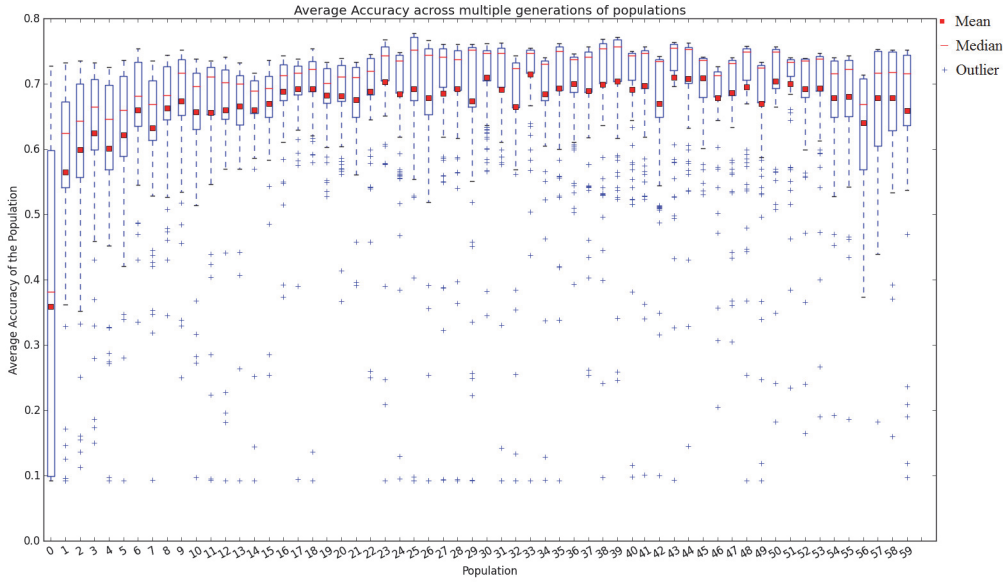

Figure 1: Distribution of fitness scores across multiple generations of populations.

classes are syntactically different. In this case, the order in which the uni-grams appear in the queries will help the classification model discriminate between the correct and incorrect class. These instances were not found in the script, therefore the genetic algorithm filtered out the high order n-gram features and converged towards uni-grams.

- For all 80 chromosomes the term weight value was 1 meaning the weighted maximum likelihood estimate should be used to calculate the probability. This result indicates that n-gram features found in the script were not unique to a single class.
- 18 chromosomes had depth 1. 62 chromosomes consisted of a depth value of 2. The use of synonyms when creating paraphrases caused the chromosomes with a depth value of 1 and 2 to outperform the chromosomes with a depth value of 0. When analyzing the script it was determined the average number of shared words per class was 4. Despite the queries within a class only sharing 4 words there was high agreement rate among the raters that queries within the classes were paraphrases. This indicates that synonyms were used when generating paraphrases and supports the selection of a depth value of 1 or 2.
- For the 80 chromosomes the median value of the word similarity threshold was 0.42 with a standard deviation of 0.18.
- For the 80 chromosomes the median value of the acceptance scalar was 0.00575 with a standard deviation of 0.0008.

The similarity between chromosomes in population 61 indicates that the genetic algorithm was capable of filtering out the irrelevant features and converging on a subset of relevant features.

The last step of the evaluation process was to evaluate the chromosomes that were identified as being fittest on a unseen test set. The average fitness score of the entire population on the unseen test set was 70.197% which was a 95% increase from the 35.929% observed in population 0. The highest fitness score of an individual chromosome on the unseen test set was 72.678%. From these results it can be stated that the genetic algorithm was able to filter out irrelevant features while selecting a subset of relevant features that allow the classification model to maximize the number of appropriate responses returned.

References

Bloodworth, T.; Cairco, L.; McClendon, J.; Hodges, L. F.; Babu, S.; Meehan, N. K.; Johnson, A.; and Ulinski, A. C. 2012. Initial evaluation of a virtual pediatric patient system. *Carolinas Women in Computing*.

Dickerson, R.; Johnsen, K.; Raij, A.; Lok, B.; Hernandez, J.; Stevens, A.; and Lind, D. 2005. Evaluating a script-based approach for simulating patient-doctor interaction. In *Proceedings of the International Conference of Human-Computer Interface Advances for Modeling and Simulation*, 79–84.

Leuski, A.; Patel, R.; Traum, D.; and Kennedy, B. 2009. Building effective question answering characters. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, 18–27. Association for Computational Linguistics.

McClendon, J. L.; Mack, N. A.; and Hodges, L. F. 2014. The use of paraphrase identification in the retrieval of appropriate responses for script based conversational agents. In *The AAAI Press*.