An Evolutionary Computing Approach Toward the Lemmatization of Textual Data[1]

Elsevier[2]

Missouri, United States

1. Abstract

The need for an increased understanding of the analysis of natural language processing is universal in scope and potential utility. In industry, linguistic data provides insight in meeting the needs of customers. Theories of language contribute to a variety of educational techniques. Recent increases in data accessibility, computational power, and innovation in strategy toward textual data manipulation have led to exponential growth and advancement in this topic area. This has contributed to increased demand for methodological development to improve upon existing theory and approaches. In this experiment, algorithmic approaches were used to evaluate relationships between words and lemmas as a proof of concept. *Keywords:* Linguistics, Evolutionary Algorithms

2010 : 00-01,99-00

2. Related Works

While grammatical and semantic properties of words are an essential aspect of all linguistic models, linguistic theory-based analyses vary greatly. Many traditional approaches result in statistical models with high dimensionality, resulting in a lack of applicability across languages or domain [1]. On the other hand, linguistic models aiming for generalization are often overly broad, also resulting in low practical use [2].

In evolutionary algorithm approaches, rule-based systems are often high quality, with lower applicability to different types of texts, while data-driven models tend to have high coverage across text sources, but lower quality of information [3]. Assessing methodology in language research is challenging. The rapid increase in demand has led to fragmentation within the field, with developers often generating highly specified or proprietary solutions without contributing to open-source tools [4] and there is no standardization in the industry in testing the efficacy of existing methods [5]. Ambiguity in language requires special consideration [3]. Due to the complexity of the task, in some cases, no method exists to assess the relationship between performance and parameters [3]. The current approach will utilize popular and freely available tools for classification as well as genetic programming as a means to evaluate word norms and relationships.

3. Methods

3.1 Data Preprocessing

Initially, a decision tree was applied to the full set as a control to determine approximate ranges for hyperparameters. Next, a subset of the dataset was selected to assess the relationship between the original cue, feature, part of speech, and the resulting lemma. A random selection (n = 4,448) of values associated with 50 cue-words appeared to most closely approximate the accuracy of the decision tree as applied to the full set (Table 1). This stage would benefit from further optimization, to ensure the highest accuracy possible in a minimal set. The categorical features were manually converted into a numerical representation for the decision tree. The unsupervised learning approaches were encoded with One Hot Encoding, a popular encoding method suitable for machine learning algorithms.

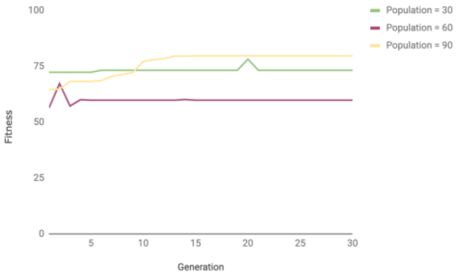
3.2 Evolutionary Programming

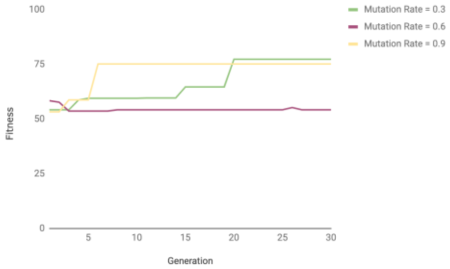
The goal of the experiment was to apply genetic algorithms toward the hyperparameters of popular machine learning algorithms and examine results in linguistic data. Sci-kit learn [6] package provides helpful documentation for the parameters, their default values, options, and ideal ranges. Crossover was performed utilizing a one-point crossover. Mutation rate was based on a randomly selected index in our chromosome. In the decision tree, survivor selection was compared, using a combination of child and actual population, as well as replacing the worst parent (Figure 2). In supervised learning, we use integer matching technique between predefined labeled outputs, which are the actual outputs, and corresponding predicted outputs to calculate fitness. If one single entity in actual output matches with respective predicted output, we perform an integer count on that. After doing integer counting for all matching predicted outputs, we converted it into percentage format simply by dividing it with the number of tested entities. Depending on that fitness, our genetic algorithm gave an understanding of how it evolves toward a more accurate solution. In unsupervised learning, two clustering score metrics homogeneity and completeness to determine a clustering result's fitness. To note their effect, the homogeneity score is calculated based on all clusters containing only data points of the same class while completeness calculates that all data points that belong to a given class belong to the same cluster.

3.3 Supervised & Unsupervised Approaches

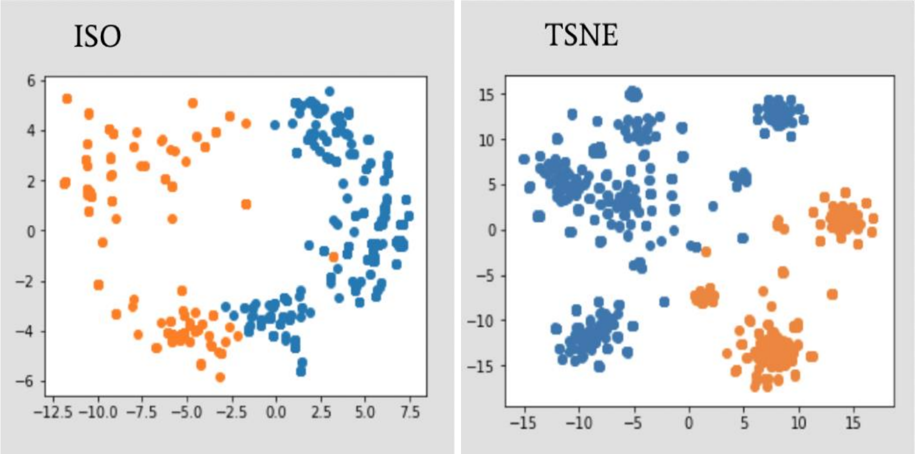
For supervised learning, a decision tree classifier was applied to create a decision based tree-like structure, with leaf node denoting the consequence of decisions. When we used the testing data set it would predict the leaf node based on the decisions on its previous root nodes. The leaf node is the outcome for each entity in the testing data set. Hyperparameters were tuned by the evolving population through generations. To gain a more complete understanding of the relations between textual data components with an unsupervised approach, Spectral Clustering, Agglomerative Clustering, and Affinity Propagation were applied. Spectral clustering is a clustering algorithm based on graph theory. Computed eigenvalue and eigenvectors contribute to clustering results. Agglomerative clustering provides an incremented "ground-up" clustering approach, revolving around varying linkage-criteria between clusters. Affinity Propagation is a useful approach for exploration, as it is an algorithm that doesn't require a desired *k*, instead, it finds exemplars that can represent a cluster. The corresponding algorithm's hyperparameters were evolved and evaluated based on the performance of the remaining population after the evolutionary process has finished.

4. Results and Discussion





While tuning the hyperparameters of decision tree classifiers, fitness increased, though default specifications were not improved on in any trial. Combining population with child population and keeping only the best individuals resulted in slower growth in fitness in comparison to replacing the worst individual in the population. Further replication supported these results. With a population of 100 chromosomes and 50 generations, both Spectral and Agglomerative Clustering converged to a population containing *k = 2* with varying hyperparameters (Figure 1). This result was confirmed with Affinity Propagation, in which exemplars resulted determine two distinct clustering results. These results are preliminary. Further theoretical consideration is needed to clarify the relationship between samples belonging to the same cluster.



5. Conclusion

In the present series of experiments, evolutionary algorithms were successful in hyperparameter optimization. Both supervised and unsupervised approaches would benefit from implementation and comparison with two-point crossover and tournament mutation. Replications would further confirm the results of the supervised approach used. In addition, more experimentation is needed to further evaluate the current methodology and theory. Bootstrapping or other boosting algorithms may be useful to determine the optimal sampling method and sample size. \cite{gleim2019practitioner} assessed lemmatization comparing fine-grained parts of speech as opposed to coarser groups, a parameter with the potential for optimization using genetic algorithms. Adjusting decision trees by applying genetic algorithms using a random forest model may increase performance or contribute to improvements in word classification, such as in producing stop-words, spell-checking, parts of speech, and semantic meaning most closely approximating natural language processing. For example, Lee, Lim, and Ahn \citep{lee2019automotive} suggest that graph-directed models may improve on tree-based models. Altering parameters further using a more complex multi-modal approach such as optimizing speed, time, and dimensionality as hyperparameter is also an opportunity to further develop tools, libraries, in the pursuit of increasing accessibility and advancing methods. Finally, replications using alternate corpora would help confirm and refine results.

References

[1] Lars Bungum and Bjorn Gamback. Evolutionary algorithms in natural language processing. In Norwegian Artificial Intelligence Symposium, Gjøvik, volume 22, 2010.

[2] Rudiger Gleim, Steffen Eger, Alexander Mehler, Tolga Uslu, Wahed Hemati, Andy Lu¨cking, Alexander Henlein, Sven Kahlsdorf, and Armin Hoenen. Practitioner’s view: A comparison and a survey of lemmatization and morphological tagging in german and latin. Journal of Language Modelling, 7(1):1–52, 2019.

[3] Sara Landset, Taghi M Khoshgoftaar, Aaron N Richter, and Tawfiq Hasanin. A survey of open source tools for machine learning with big data in the hadoop ecosystem. Journal of Big Data, 2(1):24, 2015.

[4] Jong-Hyun Lee, Sangmin Lim, and Chang Wook Ahn. Automotive ecu data-based driver’s propensity learning using evolutionary random forest. IEEE Access, 7:51899–51906, 2019.

[5] Joakim Nivre. Towards a universal grammar for natural language process- ing. In International Conference on Intelligent Text Processing and Compu- tational Linguistics, pages 3–16. Springer, 2015.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Pas- sos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit- learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.