Generating custom word embeddings for scientific names in animal taxonomy

Meet Mukadam
Department of Computer Science
Rutgers University
mm2763@scarletmail.rutgers.edu

Deep Pandya
Department of Computer Science
Rutgers University
dp976@scarletmail.rutgers.edu

Abstract

Generating knowledge from natural language data has helped us solve many artificial intelligence problems. Vector representations of words have been the driving force behind majority of the natural language processing tasks. In this paper, we discuss a novel approach for the red list classification of animal species using custom generated word embeddings. We generate two different vector embeddings using fasttext and node2vec and build a predictive model for the above mentioned classification task.

1 Introduction

Word representations have become an integral part of all natural language processing applications. Word embeddings are a representation of words in a numerical way using a set of features attempting to capture their underlying meaning. However, this feature space and its corresponding values are learnt from data and not manually labeled. These embeddings are traditionally computed by unsupervised training on a large text corpus for sufficient number of iterations that results in rich feature vectors for each word. The quality of such embedding vectors depends on the quality and size of the text corpus as well as the size of dimensionality space used to represent them. Here we propose the use of a combination of two approaches, fasttext and node2vec for Red List classification of animal species.

There are two major causes of extinction of animal and plant species. It can be because of either the habitat of the species impacted due to several reasons or the loss of genetic variation. To capture the information of the former, we have introduced the use of fasttext to bring out the subword information from a text corpus obtained through Wikipedia articles while for the latter, we have employed the use of node2vec. These two approaches give us word embeddings for scientific names involving genus and species of every animal that carry all this information across N dimensions.

2 Related Works

In recent years, various approaches to incorporate morphological information of words have been proposed. Cui et al. (2015) proposed a method to generate word embeddings for rare or unseen words by restricting morphologically similar words to have similar embeddings. Soricut and Och (2015) proposed a semi-supervised learning approach using morphologically annotated data for training word representations to encode a word's morphology. Bojanowski et al. (2017) proposed a method to generate word embeddings for character n-grams as opposed to complete words and aggregating n-gram representations to obtain word embeddings. Their method, fasttext, is completely unsupervised and works with any text corpus. Fasttext can be used to obtain word embeddings for unseen words as well as encode higher similarity between morphologically similar words. Joulin et

Preprint. Under review.

al. (2017) demonstrated the use of word embeddings generated by fasttext for fast and effective text classification achieving state-of-the-art results.

Several methods have been proposed for generating vector representations for graphical data like social networks, communication networks as well as word co-occurrence networks. Factorization, random walks and deep learning have been at the center of the majority of such methods. Grover et al. (2017) proposed a method, node2vec, based on random walks to encode graphical representations of social networks and protein sequences and achieved state-of -the-art results on benchmark multi-class classification tasks for both. Further, they also assessed the quality of vector embeddings generated using node2vec on a POS-tagger task on a large Wikipedia corpus. Goyal and Ferrara (2017) provided an extensive comparative survey on the different methods to encode vector representations for graphical data along with the results of performing several benchmarking tasks using each of those methods. Kipf and Welling (2017) introduced Variational Graph Auto Encoders, a framework for unsupervised learning on graph-structured data based on VAEs that make use of node2vec with its default parameter settings to obtain vector representations of graphical data and link prediction.

Zhang et al. (2019) proposed an approach to combine fasttext and node2vec to generate word embeddings for scientific names of drugs. They formulated a similar reasoning to encode morphology and hierarchical data in the vector representations of these drugs and evaluated their model on two benchmarking tasks for similarity measuring and two biomedical relation extraction tasks and have achieved competitive results on both.

There have been works related to prediction of extinction status of animal and plant species where spatial and morphological traits data have been used with a number of Machine Learning classifiers like Bland, Lucie et. al., Nic Lughadha et. al., Bolam, F.C. and Pelletier et. al. In all these works, the latitude and longitude of where the species are found along with characteristic traits of these species like body mass, litter size, habitat breadth, trophic level have been explicitly used as features to build classifiers. Also, these works have focused on global as well as specific regions and habitats where some of these species dwell. Nic Lughadha et. al. is a review of several of such approaches to solving the same problem.

3 Main Contribution

While some of the works use carefully collected database of species information related to geography and characteristics, we have built the entire dataset from scratch from various reliable sources. For fasttext, we needed a large training corpus to establish contextual information to incorporate the details of the habitats these species are found in. Thus, we crawled Wikipedia for articles related to every species under study. Once we retrieved the entire corpus, we cleaned and preprocessed before running the fasttext algorithm on it.

Next for node2vec, we needed the hierarchical information of animal species. Scientific names contain genus and species that bring all the similar species under one cluster. So, we captured these similarities starting from kingdom, sub-kingdom, class, etc. up to species by retrieving the entire taxonomy from itis.gov. Once we had the hierarchical data, we generated a graph to hold all the information about these relations.

Using word embeddings to capture both habitats as well as genetic information is a novel approach that we have not come across so far in any of the works related to conservation of biodiversity. Further, we use these word embeddings as feature vectors to perform a binary classification task to classify endangered vs non-endangered species. We obtain this dataset from IUCN's Red List of Threatened species which has widely been used for this task.

4 Experiment

4.1 Data Collection and generating embeddings

To generate the text corpus for training the fasttext model, we crawled Wikipedia articles for scientific names of every animal species in the animal taxonomy. In order to evaluate the quality of this text corpus, we generated tf-idf vectors for all documents (each document corresponds to a wikipedia article for a particular animal species) and calculated cosine similarity between each document. We inspected N-most similar documents for a few documents selected at random. We performed

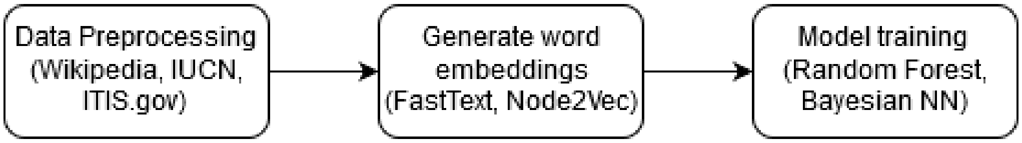
Data Preprocessing (Wikipedia, IUCN, ITIS.gov)
Generate word embeddings (FastText, Node2Vec)
Model training (Random Forest, Bayesian NN)

Figure 1: An image showing the experiment pipeline

several data cleaning and preprocessing steps and obtained the final training text corpus. We also did some analysis on this corpus. We plotted a graph for Zipf's law as shown below to validate the corpus following natural language properties. Then we ran some experiments to find the nearest documents for different species via cosine similarity and saw that we got similar species as top 5 documents. We even plotted the cosine similarity matrix for first 1000 documents. All the data cleaning, preprocessing and analysis was performed using PySpark.

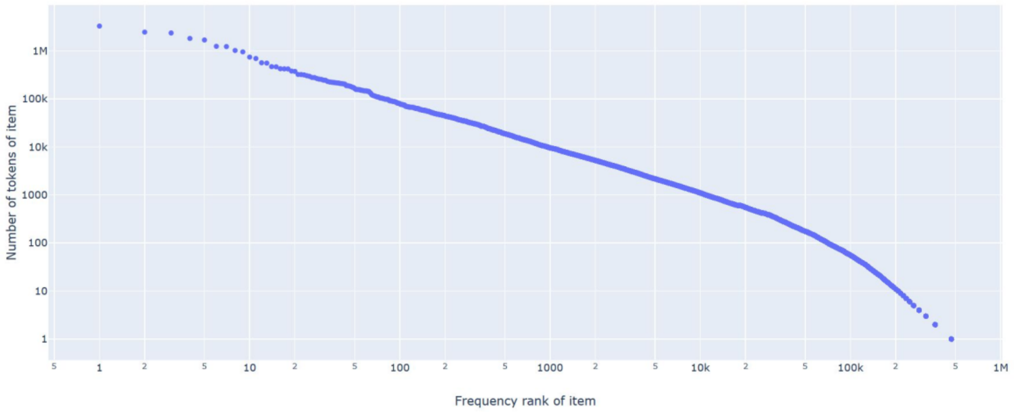

Figure 2: Zipf's law plot for the corpus

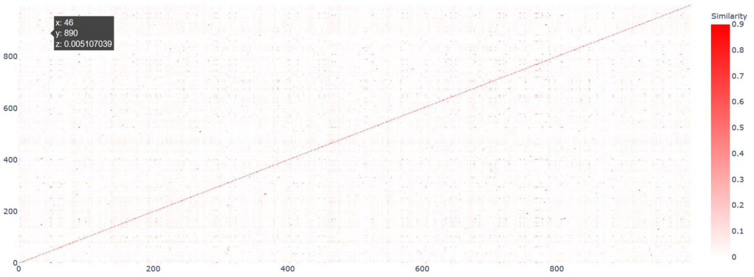

Figure 3: Document Cosine Similarity Matrix

Using fasttext on this corpus with window size of 3, we generated 200-dimensional word embeddings for each word and thus obtained word embeddings for each scientific name as well. Scientific names are a combination of genus and species, hence we take a simple average of the two to generate the final fasttext embeddings for each animal species. We also assess the quality of these word embeddings by looking at the nearest neighbors and verify that morphologically similar words are indeed being identified as similar words.

We then collected the hierarchical information of the animal taxonomy from itis.gov. The database contains hierarchy for every species using which we built a graph with nodes as every taxonomic rank and the edges depicting the relations. After generating this graph, we used node2vec to generate

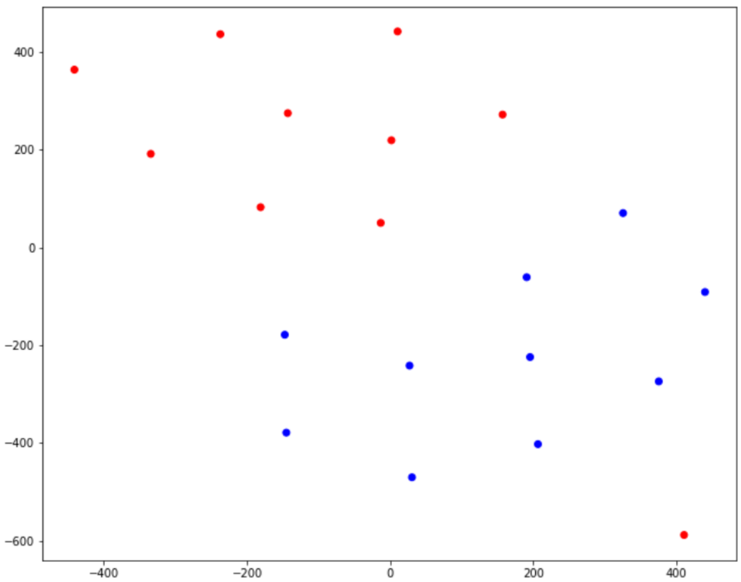

Figure 4: An image of similar animal cluster based on word embeddings. Red shows nearest neighbors for 'panthera' and blue shows nearest neighbors for 'glugea'

another 200-dimensional vector representations for each node in the graph and obtained word vectors for all species. For node2vec, we used the skip-gram model with a window size of 3 and trained it for 10 epochs with default settings of other parameters as described in Grover et. al. (2017).

We then concatenated the 200-dimensional embeddings obtained from fasttext and node2vec to obtain a 400-dimensional embedding which was later used as feature input for the machine learning classifier.

4.2 Classification

We implemented three classifiers namely vanilla neural network, Bayesian neural network and Random Forest. For the vanilla neural network, we used one hidden layer of size 32, learning rate of 0.001 with Cross Entropy loss, SGD and Adam optimizer. For the Bayesian neural network, we have used the same architecture with one hidden layer of size 32. The priors on the weights and biases have Gaussian distribution with 0 mean and unit variance. We have approximated the true posterior using Variational Bayes technique with ELBO loss, ADAM optimization and learning rate of 0.001. We used batch size of 16 and ran the model for 20 epochs after which the model converged. For both the vanilla and Bayesian neural networks, we did under-sampling to handle imbalance in the classes. Finally, we implemented a Random Forest classifier with no pruning and 250 estimators.

5 Results

Our approach is a novel one and to the best of our knowledge, we have not come across any similar work to compare our results with. There have been works related to conservation of plants Pelletier et. al. and data deficient mammals Bland et. al., but these cannot serve as benchmark for this work as the problem definition is different from ours. We ran four different classifiers on the dataset and achieved valid results. Vanilla Neural Network performed the poorest compared to others followed by the Bayesian Neural Network that was forced to predict in all situations. The reason was clearly explained when a threshold of 0.65 was placed on the confidence of the Bayesian Neural Network before making a decision. We see that the accuracy definitely increases when this confidence threshold is increased. But we also observed that the classifications into 'Unstable' class reduced because the model is always not as sure about this decision as it is about the 'Stable' class. So we see that the neural network suffers from sensitivity towards class imbalance and does not perform very well even in case of under-sampling. But Bayesian Neural Network helped perform some analysis on the results.

We see that all these shortcomings are overcome in the Random Forest classifier where it is neither sensitive to class imbalance nor struggling to make accurate decisions. We are achieving quite good results with this classifier. We have compiled the results in Table 1 as shown in this report.

6 Conclusion

In this work, we propose a novel approach of using word embeddings to capture geographical as well as genetic information for all species of the animal kingdom using a contextual training corpus and hierarchical graph dataset in the form of taxonomy. We show that uncertainty in our predictions can be modeled using Bayesian networks where the model refuses to classify when the uncertainty is fairly high. We also show that a statistical machine learning model like Random Forest performs really well and is also insensitive to the class imbalance. Through our experiments we observed that the random forest classifier works really well with the dataset we have and Bayesian approach towards neural network aided in result analysis. We have come across works on Bayesian Forests, Taddy et. al., that we think will help in enhancing the performance as well as the explanability of the classifier.

7 Tables

Table 1: Results

Model	F1 Score	MCC	Accuracy
Vanilla NN	0.578	0.059	52%
Bayesian NN (Forced to Predict)	0.667	0.064	52%
Bayesian NN (Confidence threshold = 0.65)	0.712	0.171	62%
Random Forest	0.857	0.413	78%

References

[1] Qing Cui, Bin Gao, Jiang Bian, Siyu Qiu, Hanjun Dai,and Tie-Yan Liu. 2015. KNET: A general frame-work for learning word embedding using morpholog-ical knowledge.ACM Transactions on InformationSystems, 34(1):4:1–4:25.

[2] Radu Soricut and Franz Och. 2015. Unsupervised mor-phology induction using word embeddings. In-Proc.NAACL.

[3] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information.TACL5:135–146.

[4] Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T.(2017). Bag of tricks for efficient text classification. InProc. EACL.

[5] A. Grover, J. Leskovec, node2vec: Scalable feature learning for net-works, in: Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 855–86

[6] P. Goyal and E. Ferrara. Graph embedding techniques, applications, and performance: A survey, arXiv: 1705.02801, 2017

[7] Thomas N. Kipf and Max Welling, Variational Graph Auto-Encoders, arXiv: 1611.07308, 2016

[8] Zhang, Y., Chen, Q., Yang, Z., Lin, H. & Lu, Z. BioWordVec, improving biomedical word embeddings with subword information and MeSH. Scientific Data 6, 52 (2019).

[9] Bland, Lucie Collen, Ben Orme, David Bielby, Jon. (2014). Predicting the Conservation Status of Data-Deficient Species.. Conservation biology : the journal of the Society for Conservation Biology. 29.10.1111/cobi.12372.

[10] Nic Lughadha, Eimear & Walker, Barnaby & Canteiro, Catia & Chadburn, Helen & Davis, Aaron & Hargreaves, Serene & Lucas, Eve & Schuiteman, André & Williams, Emma & Bachman, Steven & Baines, David & Barker, Amy & Budden, Andrew & Carretero, Julia & Clarkson, James & Roberts, Alex & Rivers, Malin. (2018). The use and misuse of herbarium specimens in evaluating plant extinction risks. Philosophical Transactions of The Royal Society B Biological Sciences. 374.

[11] Bolam, F.C. Addressing Uncertainty and Limited Data in Conservation Decision-Making (Doctoral Dissertation) Retrieved from https://theses.ncl.ac.uk/jspui/bitstream/10443/4389/1/Bolam%20F%202018.pdf

[12] Pelletier, Tara & Carstens, Bryan & Tank, David & Sullivan, Jack & Anahí, Espíndola. (2018). Predicting plant conservation priorities on a global scale. Proceedings of the National Academy of Sciences. 115. 201804098. 10.1073/pnas.1804098115.

[13] https://www.itis.gov/

[14] Matthew Taddy, Chun-sheng Chen, Jun Yu, and Mitch Wyle. Bayesian and empirical bayesian forests. InProceedings of the 32nd International Conference on Machine Learning (ICML-15),pages 967–976, 2015.

[15] https://github.com/paraschopra/bayesian-neural-network-mnist

[16] https://www.iucn.org/theme/species/our-work/iucn-red-list-threatened-species

[17] https://en.wikipedia.org/wiki/Zipf's_law