NATURAL LANGUAGE PROCESSING
LAB 3

Master's in Artificial Intelligence & Data Science

Done by:
MOHAMED AMINE FAKHRE-EDDINE,
Supervised by:
Lotfi EL AACHAK

19/05/2024

1 Lab report

1.1 Language Modeling - Regression

In this part of the lab, we learned about language modeling using regression, to predict the score of an answer for a specific question.

1.1.1 Data Preprocessing

We started by preprocessing the data, which consisted of tokenizing the text, removing stopwords, and converting the text to lowercase.

1.1.2 Tokenization

Tokenization is the process of splitting text into individual words or tokens. We used the 'nltk' library to tokenize the text.

1.1.3 Word Embeddings

We trained word embeddings using the 'Word2Vec' model from the 'gensim' library. Word embeddings are dense vector representations of words that capture semantic relationships between words.

So we each question with multiple answers, we trained a cbow model to have a vector representation of each word, then we aggregated the vectors of the words in the question to have a vector representation of the answers.

We used a custom function to fine-tune the model since regression models are pretty sensitive to the vector size of the word2vec model we would dbe using.

1.1.4 Regression

We used regression to predict the score of an answer given a question.

After finding the best vector size for each regression model notably 'LinearRegression', 'SVR', 'DecisionTreeRegressor', we evaluated the models using the mean squared error and the R2 score.

1.2 Language Modeling - Classification

In this part of the lab, we learned about language modeling using classification, to prediction the sentiment of a tweet.

1.2.1 Data Preprocessing

Same as the regression part, we started by extensively preprocessing the data, which consisted of removing user handles, words starting with a dollar sign, hyperlinks, hashtags, punctuations, words with 2 or fewer letters, HTML special entities, whitespace, stopwords, characters beyond the Basic Multilingual Plane (BMP) of Unicode, and converting the tweet to lowercase.

1.2.2 Tokenization

Same as the regression part, we used the ‘nltk‘ library to tokenize the text.

1.2.3 Word Embeddings

Same as the regression part, we trained word embeddings using the ‘Word2Vec‘ model from the ‘gensim‘ library.

1.2.4 Classification

We used classification to predict the sentiment of a tweet.

After finding the best vector size for each classification model notably ‘LogisticRegression‘, ‘SVC‘, ‘DecisionTreeClassifier‘, ‘AdaBoostClassifier‘, ‘LogisticRegressionCV‘ we evaluated the models using the accuracy score, and F1 score.

2 What have we learned?

Regression and classification are two common techniques used in language modeling to predict continuous and discrete values, respectively. In this lab, we applied regression to predict the score of an answer given a question and classification to predict the sentiment of a tweet. We used word embeddings to represent words as dense vectors and trained regression and classification models on these vectors. We evaluated the models using metrics such as mean squared error, R2 score, accuracy score, and F1 score.

Additionally, we learned that regression models are pretty sensitive to the vector size of the word2vec model we would be using, pretty sensitive to the smallest change in the vector size, in contrast to classification models, which are more robust to changes in the vector size, so a small change in the vector size would not affect the performance of the model that much, that’s why fine-tuning word2vec model for classification would take so much more time.

References

[1] T. Zerrouki, "pyarabic, an arabic language library for python." [Online]. Available: https://pypi.python.org/pypi/pyarabic,year={2010}

[2] ——, "Tashaphyne, arabic light stemmer," 2012. [Online]. Available: https://pypi.python.org/pypi/Tashaphyne/

[3] ——, "qalsadi, arabic mophological analyzer library for python." 2012. [Online]. Available: https://pypi.python.org/pypi/qalsadi

[4] ——, "Towards an open platform for arabic language processing," 2020.

[5] A. L. T. G. at Qatar Computing Research Institute (QCRI), "farasa, the state-of-the-art full-stack package to deal with arabic language processing." 2020. [Online]. Available: https://farasa.qcri.org/

[6] M. H. Btoush, A. Alarabeyyat, and I. Olab, "Rule based approach for arabic part of speech tagging and name entity recognition," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 6, 2016. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2016.070642

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[8] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[9] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[10] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[11] Stanfordnlp, "Github - stanfordnlp/glove: Software in c and data files for the popular glove model for distributed word representations, a.k.a. word vectors or embeddings." [Online]. Available: https://github.com/stanfordnlp/GloVe

[12] M. Toshevska, F. Stojanovska, and J. Kalajdjieski, "Comparative analysis of word embeddings for capturing word similarities," in *6th International Conference on Natural Language Processing (NATP 2020)*, ser. NATP 2020. Aircc Publishing Corporation, Apr. 2020. [Online]. Available: http://dx.doi.org/10.5121/csit.2020.100402