NATURAL LANGUAGE PROCESSING
LAB 2

Master's in Artificial Intelligence & Data Science

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

04/05/2024

1 What have we learned?

1.1 Regex

Regular expressions (regex) are a powerful tool for pattern matching and text processing. They allow us to search for and manipulate text based on patterns, making them an essential tool for any text processing task.

In this lab, we used regex to extract information from a text and generate a bill. My regex matches 4 groups, the quantity, either written as a number or words, the product name, the unit price and the dollar sign or ”dollar” word. I made a custom parser to convert the quantity to a number.

1.2 One ot encoding, Bag of Words, TF-IDF

One-hot encoding, bag of words, and TF-IDF are techniques used to convert text data into numerical vectors that can be used as input to machine learning models.

1.2.1 One-hot encoding

This technique represents each word in the vocabulary as a binary vector with a 1 at the index corresponding to the word’s position in the vocabulary and 0s elsewhere.

1.2.2 Bag of words

This technique represents each document as a vector of word counts, where each element in the vector corresponds to the count of a word in the document.

1.2.3 TF-IDF

This technique represents each document as a vector of term frequency-inverse document frequency (TF-IDF) values, which are a measure of how important a word is to a document in a collection of documents.

1.3 Word2Vec

1.3.1 Skip-gram

This model predicts the context words given a target word.

1.3.2 CBOW

This model predicts the target word given the context words.

1.3.3 GloVe and FastText

GloVe and FastText are two other word embedding techniques that learn word vectors from large text corpora.

GloVe This technique learns word vectors by factorizing the word-word co-occurrence matrix.

FastText This technique learns word vectors by considering subword information.

1.4 t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a dimensionality reduction technique that is commonly used to visualize high-dimensional data in two or three dimensions. It works by modeling the similarity between data points in the high-dimensional space and the low-dimensional space.

2 Comparaison

2.1 One-hot encoding, Bag of Words, TF-IDF

- **One-hot encoding** It is easy to implement and understand, but it can be inefficient for large vocabularies.
- **Bag of words** It is simple and efficient, but it does not capture word order or context.
- **TF-IDF** It is a more sophisticated technique that takes into account the importance of words in a document, but it can be computationally expensive.

2.2 Word2Vec, GloVe, FastText

- **Word2Vec** This model learns word representations by predicting a word's context given the word itself. In the t-SNE visualization, words with similar meanings are grouped together, indicating that the model has learned semantic relationships. However, Word2Vec does not handle out-of-vocabulary words or capture morphological relationships.
- **FastText** This model extends Word2Vec by considering subword information. In the t-SNE visualization, not only are semantically similar words grouped together, but words with similar morphological structures (e.g., same root, similar suffixes) are also close to each other. This indicates that FastText has captured both semantic and morphological relationships. Additionally, FastText can generate representations for out-of-vocabulary words by combining the vectors of their subwords, which is a significant advantage over Word2Vec.
- **GloVe** This model learns word representations by factorizing the word co-occurrence matrix. In the t-SNE visualization, semantically similar words are grouped together, similar to Word2Vec. However, GloVe does not consider subword information and cannot handle out-of-vocabulary words, similar to Word2Vec.

3 Conclusion

In this lab, we learned about various text processing techniques, including regex, one-hot encoding, bag of words, TF-IDF, Word2Vec, GloVe, FastText, and t-SNE. We compared these techniques based on their ability to capture semantic and morphological relationships between words and handle out-of-vocabulary words. We found that FastText is the most versatile model, as it can capture both semantic and morphological relationships and generate representations for out-of-vocabulary words. However, all models have successfully learned semantic relationships between words, as evidenced by the t-SNE visualization.

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