WORD2VEC HYPERPARAMETERS TUNING FOR EFFICIENT DEEP NEURAL NETWORK CLASSIFICATION

Tasneem Gamal Abdellah Moahmmed Aly, Assoc. Prof. Dr Nazlia Omar

Fakulti Teknologi dan Sains Maklumat, Universiti Kebangsaan Malaysia
43600 UKM Bangi, Selangor Malaysia.

p106220@siswa.ukm.edu.my, nazlia@ukm.edu.my

ABSTRACT

Word embedding is an emerging topic in recent years. The application of deep learning in word embedding is expanding and being used by top corporations like Google and Facebook in various applications such as search and recommendation engines. Some several techniques and libraries have been published, like Word2Vec, FastText and GloVe. Word embedding usually defines the word by its accompanying words. Researchers often choose the default hyperparameters mentioned in the original paper for word embedding by Mikolov, such as Word2Vec. The default hyperparameters have different effects on different downstream tasks. One simply cannot use the same hyperparameters for word embedding for downstream classification tasks and semantic analogy downstream tasks. The default and recommended hyperparameters work the best with specific downstream tasks of analogy and semantic meaning. Complex downstream tasks such as classification might require different hyperparameters. This research aims to study hyperparameters' impact on the downstream task of classification. This will help the researchers to identify the combination of the word embedding hyperparameters that will fit their downstream tasks. In this research, different deep neural network (DNN) classifiers were used for the downstream classification tasks such as Recurrent Neural Network, Convolution Neural Network and Long Short-Term Memory Neural Network. The DNN networks were fed by different word embedding vector spaces trained using different hyperparameters. The results show that by navigating the hyperparameters space, the downstream task of classification accuracy can increase by 4.8%. The results also show that LSTM networks are more resilient to changes in the hyperparameters. However, CNN with a specific combination of word embedding hyperparameters can achieve the highest accuracy in the classification downstream task.

Keyword: NLP, Word embedding, Word2Vec, Hyperparameter tuning, *Deep neural network* ,Classification

I. INTRODUCTION

Neural information retrieval is extensively being used in web search engines and social media data processing. Its accuracy degradation occurs due to the nature of using unsupervised machine learning techniques in processing its unstructured corpus. Sensitive scientific studies limit neural information retrieval applications due to the degradation of its accuracy. Search engines and social media frameworks recommendation systems are examples of a successful approximation of neural information retrieval.

Word Embedding is a dense representation of words in a low dimensional vector space. As a concept, it can hold the same meaning as word vectors or distributed word representation. Word embedding is a mathematical representation of words into numbers and vectors that can be processed in several vector spaces. Processing text by neural network requires a form of numerical transformation of text, and in this case, it will be numerical vectors. Works by (Mikolov et al. 2013d, 2013a, 2013b) discussed and presented promising solutions to enable the machine to learn high quality distributed vector representations to capture syntactic and semantic relations among words in the corpus.

Word embedding hyperparameters are often taken with fixated values from the literature. The hyperparameters values mention in the literature by (Mikolov et al. 2013c) are usually designed for a specific downstream task such as semantic analogies. Understanding Word2Vec fully might be overwhelming and time-consuming, specifically for researchers who are focusing on other tasks such as combining word2vec with SVM or others. Accordingly, shed light on this problem will allow the scientific community to have the means to understand how important the hyperparameters tuning task is for the downstream tasks.

Some researchers assume that the values of the same hyperparameters can be reused for other downstream tasks such as classification. Former publications (Barkan & Koenigstein 2016) and (Grbovic et al. 2016) that used Word2Vec rarely discussed the values of the hyperparameters. This often led to a decrease in the accuracy presented in the downstream task such as classification. This work demonstrates the accuracy degradation when hyperparameters tuning for Word2Vec are ignored. The main reason for the inaccuracy decline is the low quality of the word embeddings fed to the classification network (Barkan & Koenigstein 2016; Grbovic et al. 2016). This work will also demonstrate how the quality word embedding affects the overall accuracy of the downstream task.

A high-quality word embedding can be used for downstream classification tasks such as sentiment analysis. Generating high-quality word embeddings depends on the input corpus's quality and the model's hyperparameters (Caselles-Dupré et al. 2018). The definition of high-quality word embeddings varies between different downstream tasks, which means that the high-quality word embeddings designed to be used in the semantic analogy are different from the high-quality word embeddings that are designed to be used in classification. Analyzing different combinations of word embeddings hyperparameters and their effect contribute towards the classification tasks. This can be measured by the accuracy metric of the classification neural network. Thus, examining the impact of the change of hyperparameters of the word embeddings on the classification tasks is necessary

This paper consists of five (5) sections. Section I discuss the background of this study, including the issues and problems Word2Vec hyperparameters tuning. Section II discuss the Word2Vec, Hyperparameter tuning and classification. Section III elucidates the methodology used in the study. Section IV presents the findings of the work and discussion. Lastly, section V concludes the paper with a summary of the findings and recommended future work.

II. LITERATURE REVIEW

A. Overview of Word2Vec

The basic concept of word embedding is to transform the word representation into numerical values that can be processed and compared to the context and extract semantic and analogies from the corpus. Two different basic techniques that transform the corpus into numerical representations. are One-Hot representation and distributed distribution. Word2Vec is a neural network that acts as a word embedding algorithm. It turns the corpus of text into vector space that can be processed mathematically. Google labs developed Word2Vec by (Mikolov et al. 2013e) , and it is claimed to be used by Google search engine, among other algorithms. Mikolov, Chen, et al. (2013) released the code online, and the trained model of 1.6 billion words described as state-of-the-art performance for measuring syntactic and semantic word similarity.

Since Word2Vec is a neural network, it goes with the exact mechanism of input, output, and hidden layers. The layers in the neural network of Word2Vec are word embedding. The weights of the neural network of Word2Vec are initialized to random numbers. Then the network keeps adjusting the weights by going through sentence by sentence (sentences that contain the designated word) until the weights are optimized to best represent this word in the corpus. The embedding typically is the weight vectors themselves, which often coincides with the activation pattern of the hidden layer. The outstanding findings of Word2Vec are the resultant weights vectors in the hidden layer. The resultant vector of different words can be mathematically calculated against each other to answer different linguistics questions like similarity, relationship, and semantic meaning. The most common example used to describe this is "king, man, queen, woman" vectors. If the corpus is big enough, the "king" and "man" vectors should be close together.

B. Implementation of the Word2Vec

These studies were performed by prior researchers' works in different ways for applying Word2Vec hyperparameter tunning on some recommendation systems, other downstream tasks and in different languages. The word embedding is designed to analyze the corpus of text and classify it eliminates the need for users to remark anything. It is a kind of unsupervised machine learning that learns from current words and sentences without requiring interpretation. The specific abilities of word embedding are determined by how it is applied. Here are a few illustrations of how Word embedding has been used to various problems as below.

- Automated text tagging**:** Nikfarjam et al. (2015) used word embedding to extract drug response characteristics from a social media corpus, claiming an accuracy of 82 %, which is an increase over the baseline assessed.

- Recommendation Engines: Ozsoy (2016) researched the architecture of a recommendation system utilizing Word2Vec word embedding in the Foursquare check-in dataset and found significant improvement for word embedding recommendation engines.

- Machine translation: Mikolov, Le, et al. (2013)using two alternative translations for the same content to train the word embedding algorithm. Showed how monolingual data might be mapped to bilingual data. The distributed representation can generate vector space similarities and successfully translate them. Mikolov's experiment obtained an accuracy of 90% of precision for translation between English and Spanish. Chelba et al. (2013) released a new one-billion-word data set that will be used to measure statistical language modelling. This dataset has the potential to be implemented for both translation and word embedding assessments.

- Question and answers: Even though there has been much study on automating an AI agent to answer human queries, Weston et al. (2015) believe no complete system can do it yet. They claimed that having an automated AI question and answer system might be achieved by combining word embedding with an improved memory network model.

- Sentiment analysis is an excellent illustration of just how Word2Vec can be used. It is possible that categorizing user reviews will take a long time. In the classification of sentiment analysis, there are various supervised learning approaches. Using Word2Vec, on the other hand, can be a more straightforward method for sentiment analysis. The IMDB movie review dataset was researched in 2015 by a Facebook AI research group led by Mesnil et al. (2014). Several integrated machine learning techniques were used to study the IMDB movie review dataset. In their study, they studied the NB-SVM, RNN-LM, and sentence vector methods. They have made their code public to make it simple to replicate their results and increase their accountability.

III. RESEARCH MODEL AND RESEARCH QUESTIONS

The objective of this study is divided into two main sections. First, to find the best hyperparameters to generate high-quality word embeddings. Then the generated word embeddings are evaluated by testing its accuracy through downstream tasks of classification using several types of deep neural networks such as RNN, CNN, and LSTM. Comparing the result of the word embedding with the same dataset through more than one deep neural network can give a great insight into which of the neural network can achieve more accuracy rate with hyperparameter tuning and which neural network is more resilient to the hyperparameter tuning. Therefore, based on the conceptual study and the research literature, a model is designed to try hundreds of several combinations of hyperparameters of Word2Vec to obtain the highest possible quality of vector space of word embedding. The resultant vector space of word embedding will be used in the second section of the experiment. The downstream task is to classify the Amazon Custom Reviews into the predefined ranking of the reviews t. The associated factors are explained as follows.

A. Word2Vec Hyperparameters tuning on downstream task

Adewumi et al. (2020) confirmed empirically that the combination of hyperparameters is highly dependent on the downstream tasks. They have used different combinations of hyperparameters to create a high-quality vector representation that can be used to build a state-of-the-art downstream task such as classifications. They have tested their hypotheses with intrinsic and extrinsic (downstream) evaluations, including named entity recognition (NER) and sentiment analysis (SA). Outstanding findings such as high analogy scores do not necessarily correlate positively with F1 scores, and the same applies to focus on data alone.

Yildiz & Tezgider (2020) studied the hyperparameters tuning for Word2Vec to generate high-quality word embedding and vector spaces. The research focused on the parameters of the minimum word count, vector size, window size, and the number of iterations. They have also introduced two main methods: computationally more efficient than grid search and random search. They used around 300 million words. The downstream tasks were developed using deep learning classifiers. The task was to classify documents into ten different classes. The classification task was used to evaluate the quality of the generated word embedding and vector spaces. Their results show a 9% increase in the overall classification accuracy.

In addition, a study by Chamberlain et al. (2020) studied Word2Vec as an effective system mastering device that emerged from Natural Language Processing (NLP) and is now carried out in more than one domain, inclusive of recommender systems forecasting and community analysis. The researchers first elucidate the significance of hyperparameter optimization and display that unconstrained optimization yields a mean 221% development in hit charge over the default parameters. However, unconstrained optimization results in hyperparameter settings that can be very high priced and no longer viable for massive scale advice tasks. The researchers display 138% common development in hit charge with a runtime budget-restricted hyperparameter optimization.

B. Classification

the classification task takes the word embedding vector space as an input. Then train a deep neural network classifier on a downstream task classification task. The output accuracy is, of course, dependable on the Deep Neural Networks models. However, the aim is to find the discrepancy that might change the overall accuracy of changing the input word embedding layers.

C. Evaluation

The evaluation method offers a comparison of performance between the algorithms that were used in this study. Four performance metrics (Powers 2020) were used: accuracy, precision, recall, and f1-score, a commonly used metric for assessing information retrieval systems like search engines and a variety of machine learning models, particularly in natural language processing.

The case for this study is the Word2Vec hyperparameter tuning on the Amazon customer reviews dataset and evaluate the quality of the embedding via classification task. In general, the study aims to answer the following research questions:

i) RQ1: What are the best hyperparameters combination to generate high-quality word embeddings on the classification task?

ii) RQ2: What is the quality of the word embeddings via classification task performance using several types of Deep Neural Networks such as Recurrent Neural Network, Convolutional Neural Network and Long-Term Short Memory?

IV. METHODS: PARTICIPANTS AND DATA COLLECTION

This research uses the Amazon Custom Reviews dataset that contains textual data representing the customers' opinion, and it is a star rating for various products sold on Amazon.com. The dataset is composed of over 130 million reviews. This research used only 100K reviews—the link for the "Amazon Customer Reviews Dataset", 2015, https://s3.amazonaws.com/amazon-reviewspds/readme.html. The dataset contains 14 columns, the four-column that is relevant to our experiments are selected, which are the review id, review headline, review body and the star rating. Then, the four-column that is relevant to our experiments are selected, which are the review id, review headline, review body and the star rating. The five ranks are divided into two ranks. The lower rank obtains a one and two-star ranking, and the higher rank contains the three, four and five-star ranking. All punctuations and special characters, and English Stop words (Rajaraman & Ullman 2011) were removed from the text. The data was split into 75% for training and 25% for testing.

A wide range for each hyperparameter has been initialized. A range starting from 10 up to 500 values as the Dimension size, values as the window size is from 1 to 50, value as negative sampling exponent is 0.75, number epochs is from 1 to 40, minimum count value from 1 to 40 and negative sampling range from 50 to 20. This research explores both skip-gram and CBOW settings as a part of the hyperparameters navigations. The total number of hyperparameter combinations based on the hyperparameters and their ranges will result in 2,352,000,000 combinations. Each hyperparameter combination will take a long time and due to the computational resource limitation present. One thousand different combinations for the hyperparameters have been randomly chosen from the total combination for this research. At the last stage, parameters that produce the best results are saved as the best hyperparameter set. The resultant word embeddings vector space was fed to several Deep Neural Networks such as (CNN), (RNN)and (LSTM). Then, the results of each accuracy are compared to identify the best hyperparameters combination for each model. Multiple methods of measuring the accuracy of the models were used, such as accuracy, precision, recall and F1 Score for both testing and validation sets.

V. RESULTS & DISCUSSION

Based on the related work and our understanding of the hyperparameters of Gensim Word2Vec.A model has been designed to find the best combination of hyperparameters for Gensim Word2Vec to be used as an embedding layer for downstream tasks of classifications using different types of DNN such as RNN, CNN and LSTM. The models trained and then performed the hyperparameters tuning task and the data analysis using Python 3.6 Jupyter NoteBook with AWS platform as the programming environment. Furthermore, The ranges and the rationale behind its choice are listed in Table 1.

Table 1 Random range value of the hyperparameters

Hyperparameter	**Value Range**	**Hyperparameters details**
Dimension size	[10:500]	Dimension size below ten will carry a significant abstract to knowledge representation, and the accuracy will be impacted. Dimension size above 500 caused problems with memory management and processing time.
Window size	[1:50]	Window size of 1 influences the network to learn the interchangeability of entities. Window size up to 50 will allow the network to learn the relatedness analogy.
Number of Epochs	[1:40]	The greater the number of epochs, the better results. Especially if the network can prevent overfitting, such as Gensim Word2Vec. Due to the processing power limitation, the maximum value is 40
Negative sampling	[5:20]	Same as the range that Gensim Library recommends.
Skip-Gram	[0-1]	all possible variations, which are 0 and 1.
Hierarchical softmax	[0-1]	all possible variations, which are 0 and 1
Minimum count	[1:40]	If the minimum count increases above 40, the network will ignore all words with a total frequency lower than this number. It will result in minimal vocabulary size

The results were obtained with the hyperparameters tunning using Word2Vec with a classifier including CNN, RNN and LSTM. The evaluation used f-measure, accuracy, recall and precision. Key observations for the results of the F1 score, CNN achieved the highest possible F1 score of 91.1%, LSTM achieved the highest average and median F1 score of 89.9% and 90%, respectively. This indicates that LSTM is the most resilient to the variation of the hyperparameters of the embedding layers obtained by Word2Vec. RNN obtained the lowest F1 score of 86.3%. RNN is commonly known to suffer from accuracy compared to LSTM specifically. The experiment results of. the minimum, maximum, average, and median F1 Score for the testing set for different DNN networks (RNN, CNN and LSTM) indicates in Table 2

Table 2 F1 score value (min, max, average, and median) for (RNN, CNN, LSTM

	RNN	CNN	LSTM
Min	0.863	0.864	0.865
Max	0.904	0.911	0.908
Average	0.884	0.888	0.898
Median	0.885	0.893	0.9

Moreover, Figure 1 visually represents the comparison of the minimum, maximum, average and median values for F1 Score for different DNN networks (RNN, CNN and LSTM.

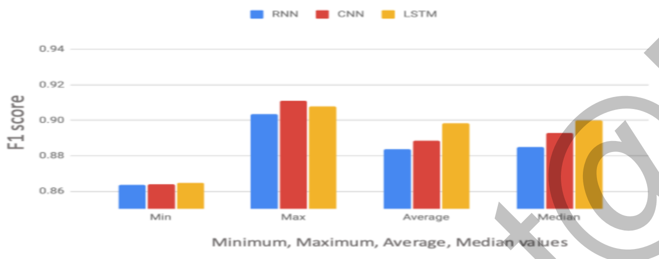

Figure 1 Visual representation of F1 score value (min, max, average and median) for (RNN, CNN, LSTM)

The top three combinations with the highest f1scoring are illustrated in Table 3. They achieved the highest accuracy of 91.11% with CNN and the highest average F1 Score on the testing set on all DNNs by 90.34%. Dimension size is greater than 200, Epoch size is larger than 18, Negative sampling value is higher than 12, and All top three combinations have Skip-Gram configured

Table 3 The best performer combinations of Gensim Word2Vec hyperparameters

Dimension Size	Window Size	Epochs	Neg-Sample	Skip-Gram	Hierarchical Softmax	Min Count	RNN	CNN	LSTM	Average
433	1	32	13	1	1	28	0.8938	0.9111	0.9053	0.9034
232	5	23	16	1	1	38	0.8978	0.9078	0.9030	0.9029
299	36	18	12	1	0	2	0.9001	0.9029	0.9035	0.9022

The worst three combinations with the lowest f1scoring are illustrated in Table 4. They achieved the lowest accuracy of 86.42% with CNN, almost identical to the lowest average F1 Score for all DNN models. Dimension size was as small as 8 for two of the lowest performer combinations. Negative sampling, on average, seems to be similar to the highest-performing combinations. Two out of three combinations have Skip-Gram configured. and Hierarchical Softmax is set to 0 in two out of three combinations

Table 4 The worst performer combinations of Gensim Word2Vec hyperparameters

Dimension Size	Window Size	Epochs	Neg-Sample	Skip-Gram	Hierarchical Softmax	Min Count	RNN	CNN	LSTM	Average
8	45	21	17	1	0	8	0.8637	0.8642	0.8648	0.8642
306	1	1	11	0	0	28	0.8642	0.8642	0.8659	0.8647
8	6	48	11	1	1	21	0.8639	0.8648	0.8680	0.8656

Although the selection of hyperparameters is totally randomized, the impact of the dimension size significantly demonstrated its impact on the average F1 Score on all DNN networks. As demonstrated in Figure 4.4, dimension size below 25 reduces the F1 score average across all networks. An exception to the case is trail number 111, which obtained an average F1 score of 86.47% "below average" to a dimension size of 306. The reason is that the window size was selected to be one, and the number of epochs was selected to be 1as well. Even Though the dimension size was 306, the minimal number of epochs and window size negatively impacted the average F1 Score.

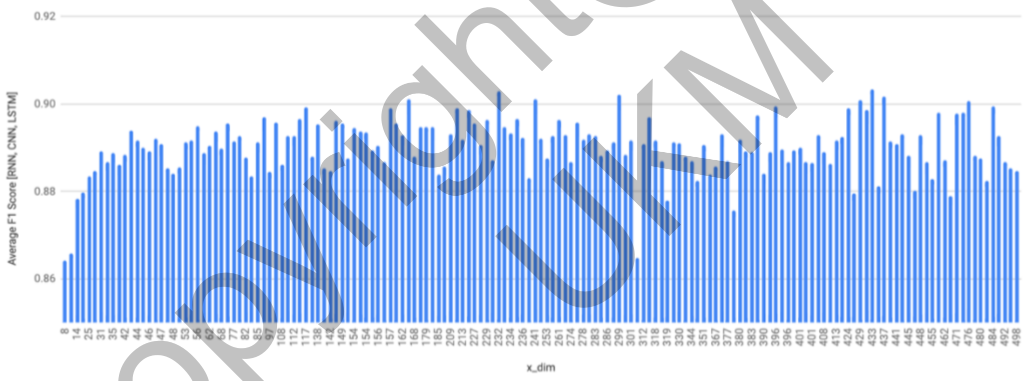

Figure 2 Dimension size impact on the overall average F1 Score

CONCLUSION

This study was designed to find the best combination of hyperparameters for Gensim Word2Vec to be used as an embedding layer for downstream tasks of classifications using different types of DNN such as RNN, CNN and LSTM. The analysis confirmed that by tuning the hyperparameters of Word2Vec and using the resultant word embedding in the embedding layer of different DNN networks, the F1 Score of the testing set could go from 86.3% to 91.1%. Therefore, by adjusting the Word2Vec hyperparameters in the experiment, one can obtain up to a 4.8% accuracy increase. Further studies can investigate the performance and comparison of Word embeddings applied to other languages and how these embeddings perform in another downstream task. In addition, studies can also be conducted on using different methods of navigating the hyperparameters space, such as grid search and Bayesian Optimization.

ACKNOWLEDGEMENT

The authors would like to thanks, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia for giving the authors an opportunity to conduct this research.

REFERENCE

Adewumi, T.P., Liwicki, F. & Liwicki, M. 2020. Word2Vec: Optimal Hyper-Parameters and Their Impact on NLP Downstream Tasks. http://arxiv.org/abs/2003.11645 [27 June 2021].

Barkan, O. & Koenigstein, N. 2016. Item2Vec: Neural Item Embedding for Collaborative Filtering. *IEEE International Workshop on Machine Learning for Signal Processing, MLSP* 2016-Novem. https://arxiv.org/abs/1603.04259v3 [30 September 2021].

Caselles-Dupré, H., Lesaint, F. & Royo-Letelier, J. 2018. Word2Vec applied to Recommendation: Hyperparameters Matter. *RecSys 2018 - 12th ACM Conference on Recommender Systems* 352–356. http://arxiv.org/abs/1804.04212 [27 June 2021].

Chamberlain, B.P., Rossi, E., Shiebler, D., Sedhain, S. & Bronstein, M.M. 2020. Tuning Word2vec for large scale recommendation systems. *RecSys 2020 - 14th ACM Conference on Recommender Systems*, pp. 732–737. Association for Computing Machinery, Inc.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P. & Robinson, T. 2014. One billion word benchmark for measuring progress in statistical language modeling. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 2635–2639. http://statmt.org/wmt11/ [27 January 2021].

Grbovic, M., Radosavljevic, V., Djuric, N., Bhamidipati, N., Savla, J., Bhagwan, V. & Sharp, D. 2016. E-commerce in Your Inbox: Product Recommendations at Scale. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 1809–1818. https://arxiv.org/abs/1606.07154v1 [30 September 2021].

Mesnil, G., Mikolov, T., Ranzato, M. & Bengio, Y. 2014. Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews. *3rd International Conference on Learning Representations, ICLR 2015 - Workshop Track Proceedings*, pp. 1–5. International Conference on Learning Representations, ICLR.

Mikolov, T., Chen, K., Corrado, G. & Dean, J. 2013a. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, pp. 1–9.

Mikolov, T., Le, Q. V. & Sutskever, I. 2013b. Exploiting Similarities among Languages for Machine

Translation. http://arxiv.org/abs/1309.4168 [27 January 2021].

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. 2013c. Distributed representations ofwords and phrases and their compositionality. *Advances in Neural Information Processing Systems*. Neural information processing systems foundation. http://arxiv.org/abs/1310.4546 [27 January 2021].

Mikolov, T., Yih, W.-T. & Zweig, G. 2013d. *Linguistic Regularities in Continuous Space Word Representations*. United States: Association for Computational Linguistics.

Mikolov, T., Yih, W.T. & Zweig, G. 2013e. Linguistic regularities in continuous spaceword representations. *NAACL HLT 2013 - 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Main Conference*, pp. 746–751. Association for Computational Linguistics. http://research.microsoft.com/en- [27 January 2021].

Nikfarjam, A., Sarker, A., O'Connor, K., Ginn, R. & Gonzalez, G. 2015. Pharmacovigilance from social media: Mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association*.

Ozsoy, M.G. 2016. From Word Embeddings to Item Recommendation. https://radimrehurek.com/gensim/models/word2vec.html [27 January 2021].

Powers, D.M.W. 2020. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. https://arxiv.org/abs/2010.16061v1 [30 September 2021].

Rajaraman, A. & Ullman, J.D. 2011. Mining of massive datasets. *Mining of Massive Datasets* 9781107015357: 1–315.

Weston, J., Bordes, A., Chopra, S., Rush, A.M., Van Merriënboer, B., Joulin, A. & Mikolov, T. 2016. Towards AI-complete question answering: A set of prerequisite toy tasks. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. https://www.researchgate.net/publication/272522139_Towards_AI-Complete_Question_Answering_A_Set_of_Prerequisite_Toy_Tasks [27 January 2021].

Yildiz, B. & Tezgider, M. 2020. Learning Quality Improved Word Embedding with Assessment of Hyperparameters. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*vol. 11997 LNCS, , pp. 506–518. Springer.