Constraining Weighted Word Co-occurrence Frequencies in Word Embeddings

Paula Lauren
Department of Mathematics and Computer Science
Lawrence Technological University
Southfield, MI, USA
plauren@ltu.edu

2021 IEEE International Conference on Big Data (Big Data) | 978-1-6654-3902-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigData52589.2021.9671892

***Abstract*—Weighted word co-occurrence frequencies are considered the bedrock of word embeddings. Also known as a low-dimensional numerical representation, word embeddings capture word pair frequencies extracted from a corpus in an unsupervised manner. The rendering of word embeddings can be considered a two-step process with the first step involving the building of the word context matrix then using a matrix factorization method to reduce the dimensionality. In this research study, word embeddings are constructed from scratch in building the word context matrix and Truncated Singular Value Decomposition is applied to the matrix. Five experimental values are defined for constraining the frequency weights in the word embeddings, which are then evaluated in word similarity and sequence labeling tasks with results reported. The word similarity task shows comparable results across all experimental constraint values. Overall comparable results are also achieved in the sequence labeling task. The experiments conducted in this study have shown promising results, which will entail future work with evaluation on other tasks.**

***Index Terms*—Word Embeddings; Word Similarity; Text Classification; Word Co-occurrence Matrix; Word Context Matrix**

I. Introduction

Word embeddings have been utilized in various Natural Language Processing (NLP) tasks such as word similarity, sentence similarity, analogy, text classification, speech recognition, text summarization, and question-answering systems [1]–[7]. Several neural network methods used in NLP have been modified to include a dedicated embedding layer specifically for containing word embeddings, from convolutional neural networks [8] to transformer neural networks [9]. Word embeddings have also been tailored with applicability towards disparate domains from music [10] to genetics [11].

Weighted word co-occurrence frequencies are considered the bedrock of word embeddings, which capture word pair frequencies extracted from a corpus and rendered using a low-dimensional numerical representation. Word embeddings are contained in a two-dimensional matrix consisting of these word pair frequencies where each row or column represents a word vector. The generation of word embeddings can use a prediction-based or a count-based approach [12]. Prediction-based models typically utilize a neural network in the rendering of word embeddings and count-based models typically use statistical methods. Although, there has been some research into tensor-based word embeddings [13]–[15] that extend beyond one-dimensional word vectors, the focus of this study is on one-dimensional word vectors and their two-dimensional matrix origin.

Word2Vec and Global Vectors (GloVe) have been the predominate approaches for creating word embeddings, where as the former takes a prediction-based approach and the latter takes a count-based approach. Both Word2Vec and GloVe have also made available pretrained word embeddings, which are trained on tens of billions of tokens, also known as words [16], [17]. High frequency words are addressed in Word2Vec to differentiate between rare and frequent words using a subsampling approach on the training set. The method was heuristically determined to subsample words whose frequency is greater than a threshold while maintaining the frequency ranking. The GloVe model defines a parameter for controlling the max frequency. It appears that the frequency threshold in both models are done a posteriori, which increases the time to render the word embeddings. In addition, the time for generating word embeddings for GloVe on a single machine had taken about 85 minutes for a 400,0000 word vocabulary trained on six billion words [17].

This research paper focuses on the frequency weights for word co-occurrences by imposing various experimental constraints a priori on the frequency weights and evaluating the results using both word similarity and sequence labeling as the classification task. The motivation is to assess the impact that word frequency has on the usefulness of word embeddings in the experimental tasks. Furthermore, the time to generate the word embeddings is also captured for each constraint imposed on the rendering of the word embeddings. The organization of this paper is as follows: Section II is the layout of the experimental design for the experiments where five constraint values are defined on the two aforementioned tasks. Section III is where the results and evaluation of the word embeddings using the experimental contraint values are provided along with evaluation. Future work is discussed in Section IV with the conclusion in Section V.

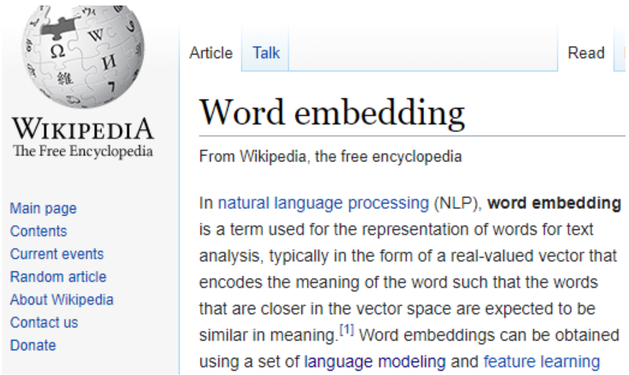

Fig. 1. Snippet of a Wikipedia article.

TABLE I
SUMMARY STATISTICS FOR WIKIPEDIA DATASET.

Measure	Words
Total number of articles	20,718
Average number of words	3,165
Minimum number of words	50
Number of words in 25^{th} percentile	719
Number of words in 50^{th} percentile	2,054
Number of words in 75^{th} percentile	4,447
Maximum number of words	36,191

II. EXPERIMENTAL DESIGN

In this section a description of the experiments are provided for assessing various constraints on the co-occurrence weight frequencies in generating word embeddings. A description of the datasets are provided that are utilized in the experiments. Constraints for the word embeddings are defined, which are then subsequently applied to the tasks of word similarity and sequence labeling.

A. Datasets and Data Preprocessing

Two datasets are used in this study for the word similarity task and the sequence labeling task and each are described according to each task.

*1) **Dataset for the Word Similarity Task**:* The dataset for generating the word embeddings in assessing word similarity utilized Wikipedia data, which has been made publicly available via a Wikipedia data collection website[1]. For this study the *enwiki-latest-pages-meta-current1.xml-p1p41242* file had been downloaded containing a subset of the wikipedia articles on August 14th, 2021, approximately 270 MB. This dataset contains 20,718 articles, where each article corresponds to a Wikipedia page as illustrated in Figure 1.

Summary statistics are given in Table I providing various statistical measures for describing the number of words from the Wikipedia articles. The smallest article contains 50 words and the largest article has 36,191 words. The average article length contains 3,165 words. Data preprocessing entailed punctuation removal, lowercase conversion, and tokenization.

[1]https://dumps.wikimedia.org/enwiki/latest/

TABLE II
IOB REPRESENTATION OF AN EXAMPLE UTTERANCE FROM THE ATIS DATASET, TOKENIZED FROM THE SENTENCE.

Sentence Tokens	Named Entity
show	O
me	O
the	O
first	B-class type
class	I-class type
fares	O
from	O
phoenix	B-from location city name
to	O
detroit	B-to location city name

TABLE III
SUMMARY STATISTICS FOR ATIS DATASET (TRAINING).

Measure	Words
Total number of utterances	4,978
Average number of words	12
Minimum number of words	1
Number of words in 25^{th} percentile	8
Number of words in 50^{th} percentile	11
Number of words in 75^{th} percentile	14
Maximum number of words	46

*2) **Dataset for the Sequence Labeling Task**:* The dataset used for the Sequence Labeling task is the Air Travel Information Service (ATIS)[2], considered the most widely used corpus benchmark by the Natural Language Understanding community [18]–[20]. This dataset contains the textual representation of spoken utterances pertaining to flight reservations. There are a total of 127 named entities or classes contained in the ATIS dataset. One sentence can contain one or more entity types or classes. This dataset uses the IOB [21] format where the word is prefixed B along with a specific entity type, if the word is the beginning of an entity. An I prefix is given if the word is inside the subsequent word of a named entity. An O prefix is assigned to remaining words outside of the detection task. Table II contains an example sentence or utterance from the ATIS training dataset along with the associated named entity, illustrating the IOB representation.

Summary statistics are given in Table III providing various statistical measures for describing the number of words from the ATIS dataset. The average length of the utterances for the training set is around 12 words. The dataset reserved for training has 4,978 sentences and the dataset reserved for testing contains 893 sentences.

B. Word Embeddings

The process for creating word embeddings can be viewed as a two-step process. The first step entails the building of the word context matrix, which is also referred to as the word co-occurrence matrix. The second step involves the factorization of the word context matrix into manageable dimensional word vectors.

[2]http://lisaweb.iro.umontreal.ca/transfert/lisa/users/mesnilgr/atis/

$$\begin{array}{cc} & \begin{array}{cccc} \textit{artwork} & \textit{lake} & \textit{michigan} & \textit{sun} \end{array} \\ \mathbf{X} = & \left(\begin{array}{cccc} x_{(1,1)} & x_{(1,2)} & x_{(1,3)} & x_{(1,4)} \\ x_{(2,1)} & x_{(2,2)} & x_{(2,3)} & x_{(2,4)} \\ x_{(3,1)} & x_{(3,2)} & x_{(3,3)} & x_{(3,4)} \\ x_{(4,1)} & x_{(4,2)} & x_{(4,3)} & x_{(4,4)} \end{array}\right) \begin{array}{l} \textit{artwork} \\ \textit{lake} \\ \textit{michigan} \\ \textit{sun} \end{array} \end{array}$$

Fig. 2. Illustration of a word context matrix built from a four word vocabulary where each element of the matrix denotes a weighted word pair frequency.

*1) **Word Context Matrix Construction**:* The algorithm defined in previous work for building the word context matrix has been utilized [22], [23]. The two key parameters in the rendering of the word context matrix are the *minimum word count* and *context window size*. The former is the minimum number of times the word must be in the corpus for inclusion in vocabulary V. The latter is the number of words to the left and right of the center word. It is the weighted sum of these word pairs within the *context window size* that construct the word context matrix. The weight is computed by $\frac{1}{(i+1)}$ where i is the distance between the center word and the surrounding words within the predefined context window. Summation involves keeping an updated sum of the word pairs encountered from the dataset, which are stored as an element in the matrix as in $x_{m,n} \leftarrow x_{m,n} + \frac{1}{(i+1)}$ where *m*, *n* denotes indices in matrix. The matrix dimensions *M* and *N* are both determined from $|V|$, the size of the vocabulary. To elucidate, Figure 2 contains a `4 x 4` matrix defined as X to represent an illustration of a word context matrix constructed from a four word vocabulary with words captured from the sentence: *We are viewing artwork by the lake in the Michigan sun.* The vocabulary (consisting of four words) define the rows and columns of the word context matrix. The word pair weight for (*artwork, lake*) would be 0.25 for the weighted frequency (i.e. `1/(i + 1) or 1/(3+1) = 0.25`) in both $x_{(1,2)}$ and $x_{(2,1)}$ elements of the example matrix in Figure 2, if stopwords are not removed. These word pair weighted frequencies will grow quite large for frequent word pairs encountered during the building of the word context matrix.

The largest word pair frequency in the Wikipedia dataset used for this study is almost 1.5 million. This research imposes constraints on these frequencies in order to ascertain any impact that constraints may have using various experimental values.

For the experiments, the constraints on the weighted word co-occurrence frequencies are as follows:

1) constraint value: `0.1`
2) constraint value: `1.0`
3) constraint value: `10.0`
4) constraint value: `100.0`
5) constraint value: unlimited `(no constraint)`

Five experiments were conducted using these constraint values on each of the two tasks resulting in an overall of 10 experiments. In constructing the word context matrix the *minimum word count* is set to 5 resulting in a vocabulary size consisting of 176,749 words on the Wikipedia dataset.

*2) **Word Context Matrix Decomposition**:* The word context matrix for the Wikipedia dataset resulted in a `176,749 by 176,749` matrix, containing of mostly sparse values. Working with `176,749` dimensional word vectors would be computationally intensive. Singular value decomposition (SVD) is a matrix factorization method used for dimensionality reduction and a modification to SVD is Truncated SVD [24]. The original SVD equation is in Equation (1)(a) and Truncated SVD is in Equation (1)(b),

$$(a) \quad X = U\Sigma V^T \qquad (b) \quad X \approx \tilde{U}\tilde{\Sigma}\tilde{V}^T \tag{1}$$

where U and V are regarded as unitary matrices with orthonormal columns and Σ is a matrix with the diagonal containing non-negative entries and zeros off the diagonal. Truncated SVD is no longer an exact decomposition of the original matrix X where tilde (˜) denotes the truncated matrices in Equation (1)(b). Truncated SVD provides a close approximation, which has been shown to be sufficient [24]. In this study Truncated SVD is applied to the word context matrix using the Sklearn library for decomposition in Python[3]. Specifically, the TruncatedSVD function is utilized where the key parameter in the function is *n_components*, which denotes the reduction of dimensionality. For this study, the parameter is set to 300 dimensions, which denotes the *word vector size* in the context of word embeddings. The application of SVD to the word context matrix is what gives the word embeddings their low-dimensional numerical representation transforming the original word context matrix with `176,749 by 176,749` dimensions into `176,749 by 300` dimensions resulting in `300` dimensional word vectors. Note that the application of SVD to a term document matrix is referred to as Latent Semantic Analysis [25].

III. Results and Evaluation

The experimental constraints applied to the rendering of the word embeddings are assessed in the word similarity task and the sequence labeling task using each of the constraints stated in the previous section.

A. Word Similarity Task

For evaluating word similarity, the cosine similarity is used to compare sample query words from the word embeddings created in the previous section using each of the experimental constraint values from the Wikipedia dataset. The cosine similarity is defined in Equation 2:

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}.\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} = \frac{\sum_{i=1}^{n} \mathbf{x}_i \mathbf{y}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{x}_i)^2}\sqrt{\sum_{i=1}^{n} (\mathbf{y}_i)^2}} \tag{2}$$

where $\mathbf{x}$ and $\mathbf{y}$ denotes word vectors. The results from the five experiments in the word similarity task are shown in Table IV-Table VIII, where each table reports on the results of the experimental constraint values. The top three most similar words are reported from a sample of query words based on the cosine similarity.

[3]https://scikitlearn.org

TABLE IV
WORD EMBEDDINGS WITH CONSTRAINT 0.1 DURING WORD CONTEXT MATRIX CONSTRUCTION.

query word	1st sim. word	2nd sim. word	3rd sim. word
artwork	illustrations	drawings	paintings
lake	valley	bay	mountain
literature	contemporary	literary	writing
michigan	ohio	illinois	pennsylvania
phrases	vocabulary	meanings	expressions
piano	guitar	orchestra	jazz
sun	moon	earth	sky
tennis	hockey	basketball	rugby

TABLE V
WORD EMBEDDINGS WITH CONSTRAINT 1.0 DURING WORD CONTEXT MATRIX CONSTRUCTION.

query word	1st sim. word	2nd sim. word	3rd sim. word
artwork	drawings	illustrations	prints
lake	valley	mountain	river
literature	literary	contemporary	writing
michigan	ohio	illinois	pennsylvania
phrases	meanings	rhyme	sentences
piano	orchestra	violin	guitar
sun	moon	stars	planet
tennis	hockey	basketball	rugby

TABLE VI
WORD EMBEDDINGS WITH CONSTRAINT 10.0 DURING WORD CONTEXT MATRIX CONSTRUCTION..

Query Word	1st sim. word	2nd sim. word	3rd sim. word
artwork	drawings	illustrations	photographs
lake	valley	mountain	river
literature	contemporary	literary	historical
michigan	ohio	illinois	pennsylvania
phrases	meanings	verbal	grammatical
piano	orchestra	guitar	solo
sun	moon	earth	stars
tennis	hockey	basketball	rugby

TABLE VII
WORD EMBEDDINGS WITH CONSTRAINT 100.0 DURING WORD CONTEXT MATRIX CONSTRUCTION.

Query Word	1st sim. word	2nd sim. word	3rd sim. word
artwork	drawings	illustrations	sketches
lake	river	valley	mountain
literature	literary	contemporary	poetry
michigan	illinois	ohio	wisconsin
phrases	expressions	verbs	nouns
piano	orchestra	violin	guitar
sun	moon	earth	planet
tennis	basketball	hockey	rugby

The word embeddings for all five experiments in the word similarity task are remarkably comparable. The results seems to suggest that it may not be the quantity of word frequencies that are significant. Reported in Table IX is the time to compute the word embeddings with the various experimental constraint values. The time to generate the word embeddings from the Wikipedia data is reduced significantly when comparing no constraint at around 41 minutes to constraint value `0.10` at around 28 minutes.

B. Sequence Labeling Task

The sequence labeling task is a classification problem and utilizes the same approach from previous work using a Recurrent Neural Network (RNN) [22]. The word embeddings generated using the five experimental constraint values for this task are the embedding layer to the RNN, which is an addition to the original RNN architecture for use in NLP tasks. Epochs are set at 10, this entails 10 passes through the training set. A dropout layer set at 0.10, which has a regularization effect by randomly dropping hidden neurons to prevent overfitting during training [26]. Evaluation is done using the standard evaluation measures of Precision, Recall, and F_1-Score as defined in Equations 3-5.

$$Precision = \frac{True\,Positive}{True\,Positive + False\,Positive} \tag{3}$$

$$Recall = \frac{True\,Positive}{True\,Positive + False\,Negative} \tag{4}$$

$$F_1 = 2\,.\,\frac{Precision\,.\,Recall}{Precision + Recall} \tag{5}$$

Precision, Recall and the F_1-Score are reported for 10 executions on the test set using an RNN for the Sequence Labeling task. Table X reports the results in this task using the mean and standard deviation across 10 iterations on the test set using the evaluation measures defined in Equations 3-5 for each of the five experiments. The results show that the experimental constraint value of `100.0` has a slightly better average in terms of the F_1-Score in comparison with the other four experiments on the sequential labeling task. Interestingly, the greatest variation is shown with the experiment having constraint `1.0` on this task, showing greater variation in each iteration. Experimental constraint value `0.1` performed slightly better in mean and standard deviation over constraint value `1.0`. The experiment with no constraint resulted in the lowest variation averaged across the 10 iterations on this task.

IV. FUTURE WORK

The word similarity task using query words shows comparable results across all of the experimental constraint values `[0.1, 1.0, 10.0, 100.0, no constraint]`. These results seem to suggest that word pair frequencies may not be the significant factor in generating meaningful word embeddings. Future work will explore this further by investigating word pair variety along with word pair frequency as well as the inclusion of additional tasks and datasets.

In the sequence labeling task, the experimental constraint `100.0` had a slightly better result in F_1-Score. Though, it is interesting that the experimental constraint value `0.1` performed slightly better than experimental constraint `1.0` in mean and standard deviation across 10 iterations. Future work will also include additional tasks with these experimental constraint values to fully assess the usefulness of applying a constraint and its impact on accuracy and computational efficiency.

TABLE VIII

WORD EMBEDDINGS WITH NO CONSTRAINT DURING WORD CONTEXT MATRIX CONSTRUCTION.

Query Word	1st sim. word	2nd sim. word	3rd sim. word
artwork	illustrations	paintings	drawings
lake	river	valley	mountain
literature	literary	poetry	contemporary
michigan	illinois	wisconsin	oregon
phrases	expressions	words	sounds
piano	violin	orchestra	guitar
sun	moon	planet	sky
tennis	basketball	rugby	baseball

TABLE IX

TIME (IN MINUTES) TO COMPUTE WORD EMBEDDINGS USING THE EXPERIMENTAL CONSTRAINTS.

Constraint	Time (in minutes)
constraint@0.10	28.02
constraint@1.0	31.20
constraint@10.0	35.58
constraint@100.0	38.41
no constraint	41.32

V. CONCLUSION

This research study involved imposing constraints on the weighted co-occurrence word frequencies in word embeddings to assess the impact to performance, in word similarity and sequence labeling tasks. As stated, the word similarity task using query words showed comparable results across all of the experimental constraint values `[0.1, 1.0, 10.0, 100.0, no constraint]`. In the sequence labeling task, the experimental constraint `100.0` had a slightly better result in F_1-Score but overall the results are comparable.

In the original GloVe paper, word embeddings generation took about 85 minutes for a 400,0000 word vocabulary on six billion words on a single machine [17]. In this study, the experimental contraint of 0.1 in generating word embeddings for approximately 176,000 words had taken 21 minutes on a comparable machine. Achieving comparable results requiring less computational resources also shows promising results that are worth exploring further.

TABLE X

COMPARISON OF RESULTS ON THE SEQUENCE LABELING TASK USING CONSTRAINT AT 0.1 (C@0.1), CONSTRAINT AT 1.0 (C@1.0), CONSTRAINT AT 10.0 (C AT 10.0), CONSTRAINT AT 100.0 (C@100.0) AND NO CONSTRAINT. THE MEAN AND STANDARD DEVIATION ARE REPORTED ON THE TEST SET.

Constraint	Precision $\mu \pm \sigma$	Recall $\mu \pm \sigma$	F_1-Score $\mu \pm \sigma$
C@0.1	92.463±0.402	91.484±0.787	91.970±0.585
C@1.0	92.545±0.517	91.374±0.704	91.955±0.602
C@10.0	92.527±0.498	91.518±0.580	92.020±0.523
C@100.0	92.513±0.493	91.570±0.667	92.117±0.460
no constraint	92.367±0.297	91.492±0.305	91.927±0.298

REFERENCES

[1] I. Iacobacci, M. T. Pilehvar, and R. Navigli, "Sensembed: Learning sense embeddings for word and relational similarity," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015, pp. 95–105.

[2] Z. Quan, Z.-J. Wang, Y. Le, B. Yao, K. Li, and J. Yin, "An efficient framework for sentence similarity modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 4, pp. 853–865, 2019.

[3] N. Hartmann, E. Fonseca, C. Shulby, M. Treviso, J. Rodrigues, and S. Aluisio, "Portuguese word embeddings: Evaluating on word analogies and natural language tasks," *arXiv preprint arXiv:1708.06025*, 2017.

[4] P. Lauren, G. Qu, F. Zhang, and A. Lendasse, "Clinical narrative classification using discriminant word embeddings with elm," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 2931–2938.

[5] T. Stafylakis and G. Tzimiropoulos, "Deep word embeddings for visual speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4974–4978.

[6] G. Rossiello, P. Basile, and G. Semeraro, "Centroid-based text summarization through compositionality of word embeddings," in *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, 2017, pp. 12–21.

[7] D. Teney, P. Anderson, X. He, and A. Van Den Hengel, "Tips and tricks for visual question answering: Learnings from the 2017 challenge," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4223–4232.

[8] A. Jacovi, O. S. Shalom, and Y. Goldberg, "Understanding convolutional neural networks for text classification," *arXiv preprint arXiv:1809.08037*, 2018.

[9] A. M. Braşoveanu and R. Andonie, "Visualizing transformers for nlp: a brief survey," in *2020 24th International Conference Information Visualisation (IV)*. IEEE, 2020, pp. 270–279.

[10] S. Garcia-Valencia, "Embeddings as representation for symbolic music," *arXiv preprint arXiv:2005.09406*, 2020.

[11] P. Mitra, T. Pijnenburg, and V. Sazonau, "Discovering gene-disease associations with biomedical word embeddings," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2020, pp. 163–170.

[12] F. Almeida and G. Xexéo, "Word embeddings: A survey," *arXiv preprint arXiv:1901.09069*, 2019.

[13] Z. Rahimi and M. M. Homayounpour, "Tenssent: a tensor based sentimental word embedding method," *Applied Intelligence*, pp. 1–16, 2021.

[14] E. Bailey and S. Aeron, "Word embeddings via tensor factorization," *arXiv preprint arXiv:1704.02686*, 2017.

[15] D. Milajevs, D. Kartsaklis, M. Sadrzadeh, and M. Purver, "Evaluating neural word representations in tensor-based compositional settings," *arXiv preprint arXiv:1408.6179*, 2014.

[16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[17] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[18] C. T. Hemphill, J. J. Godfrey, G. R. Doddington *et al.*, "The atis spoken language systems pilot corpus," *In Proceedings of the DARPA speech and natural language workshop*, 1990.

[19] D. A. Dahl, M. Bates, M. Brown, W. Fisher, K. Hunicke-Smith, D. Pallett, C. Pao, A. Rudnicky, and E. Shriberg, "Expanding the scope of the atis task: The atis-3 corpus," in *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, 1994, pp. 43–48.

[20] G. Tur, D. Hakkani-Tür, and L. Heck, "What is left to be understood in atis?" in *Spoken Language Technology Workshop (SLT), 2010 IEEE*. IEEE, 2010, pp. 19–24.

[21] L. A. Ramshaw and M. P. Marcus, "Text chunking using transformation-based learning," in *Natural language processing using very large corpora*. Springer, 1999, pp. 157–176.

[22] P. Lauren, G. Qu, J. Yang, P. Watta, G.-B. Huang, and A. Lendasse, "Generating word embeddings from an extreme learning machine for

sentiment analysis and sequence labeling tasks," *Cognitive Computation*, vol. 10, no. 4, pp. 625–638, 2018.

[23] P. Lauren, G. Qu, G.-B. Huang, P. Watta, and A. Lendasse, "A low-dimensional vector representation for words using an extreme learning machine," in *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 1817–1822.

[24] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[25] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American society for information science*, vol. 41, no. 6, pp. 391–407, 1990.

[26] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.