Studying the Dependence of Embedding Representations on the Target of NLP Tasks

Bárbara Stéphanie Neves Oliveira[1], **Ticiana L. Coelho da Silva**[1], **José A. F. de Macêdo**[1]

[1]Insight Data Science Lab
Universidade Federal do Ceará (UFC) – Fortaleza, CE – Brazil

{barbaraneves,ticianalc,jose.macedo}@insightlab.ufc.br

Abstract. *In many human languages, linguistic units represent text structure. Vector semantics is used in NLP to represent these units, known as embeddings. Evaluating the learned representations is crucial for identifying critical differences between the diverse existing embedding models in task-specific selection. However, the evaluation process is complex, with two approaches: intrinsic and extrinsic. While useful, aggregated evaluations often lack consistency due to result misalignment. This work investigates the dependencies and correlations between embeddings and NLP tasks. The goal is how to initially verify if the embeddings' dimensions (i.e., features) depend on the final task. The study then explores two research questions and presents findings from experiments.*

1. Introduction

In many human languages, most information about the structure of texts can be represented in the form of linguistic units. Understanding how to learn textual representations using Deep Learning techniques is a crucial area of research in Natural Language Processing (NLP) [Jurafsky and Martin 2018, Oliveira et al. 2022]. This focus has given rise to various architectures that aim to model words or other linguistic units such as characters, sentences, or documents.

Embeddings are a fundamental concept in NLP that serve as a form of textual representation. They are numerical vectors that encode both the meaning and contextual information of linguistic units within a given language. Many methods have been developed to generate embeddings, from more straightforward approaches to sophisticated techniques [Torregrossa et al. 2021, Oliveira et al. 2022].

The field of NLP has yet to converge on a universal embedding method and scale it sufficiently to provide state-of-the-art results on all tasks [Ignat et al. 2023, Muennighoff et al. 2022]. Consequently, constructing effective NLP pipelines with high-quality input representations remains challenging, especially with abundantly available techniques. This leads to confusion about which model provides practitioners with the best performance for their embedding use case. Thus, assessing the learned representations is vital in identifying the critical distinctions between various embedding models, enabling the selection of the most suitable one for a specific task [Boggust et al. 2022, Bakarov 2018].

Evaluating embeddings involves two primary approaches: extrinsic and intrinsic. While extrinsic evaluation guarantees practical performance, intrinsic evaluation offers insights into the inherent quality of embeddings [Jurafsky and Martin 2018]. However, further advancements are needed to enhance the evaluation and comparison process, bridging existing gaps and unlocking the full potential of these powerful language representation models [Schnabel et al. 2015, Bakarov 2018, Torregrossa et al. 2021].

In recent years, researchers have recognized the importance of addressing the preliminary verification of embeddings before utilizing them to represent a corpus or corpora in an NLP task [Boggust et al. 2022, Muennighoff et al. 2022]. This research then tackles the fundamental challenge of measuring (i.e., with heuristics or numerical measures) the dependencies and correlations between the input textual representations and the ultimate objective of an NLP task.

Contributions. Overall, the main contributions of this paper can be summarized as follows:

- Investigate the crucial step of examining whether the learned vectors, used as features, are relevant to the final task, ensuring high-quality representations.
- Present two main research questions to guide the study, providing detailed discussions, experimental setups, and results for each one.
- Conduct extensive experiments exploring whether numerical measures can determine the dependence between input embeddings and their suitability for a specific NLP task.

2. Related Work

To address the challenge of verifying input representations for NLP tasks, research works mainly focuses on three topics, presented below. This work addresses the first topic, while the remaining topics are summarized to provide an overview of the current state.

Explainability and interpretability. General techniques use different tools to understand model predictions, feature importance, and decision-making processes [Hamilton et al. 2016, Ribeiro et al. 2016, Shrikumar et al. 2017, Carter et al. 2019]. Unlike other methods, this work compares embeddings learned by different models using a global measure, considering that internal representations can vary.

Visual embedding techniques and tools. To reason about and interpret the learned representations, [Heimerl and Gleicher 2018], [Liu et al. 2019b], and [Boggust et al. 2022] propose interactive or static systems for exploring embeddings via direct projection manipulation, interactively filtering, and reconfiguring visual forms.

Methods for comparing embedding spaces. To compare vector spaces, research works perform alignment through linear transformation [Chen et al. 2018] or nearest neighbors and co-occurrences over time [Heimerl and Gleicher 2018, Wang et al. 2018b], relationship analysis between node metrics and graph embeddings [Li et al. 2018], and evaluation of vector consistency across latent embedding spaces [Liu et al. 2019b, Boggust et al. 2022].

3. Research Questions and Discussion

Before delving into the experiments, the main research questions that guide this study are:

RQ1 Can heuristics or numerical measures determine the dependence between the input embeddings and their suitability for a particular NLP task?

Context. Some model performances can degrade when including input features irrelevant to the target labels. Typically, feature selection methods are intended to reduce the number of input features to those considered most beneficial based on statistical tests [Butcher and Smith 2020].

Quantitative investigation. This research question aims to analyze the quality of pre-trained input embeddings[1] by applying an existing feature selection measure to different types of representations and corpora. The purpose is not to select the best dimensions for a specific task but to identify which embedding approach has more dimensions with high scores indicating strong dependence between input and output. The experiments focus on embedding representations for the training set.

RQ2 To what extent does the model developed to solve an NLP task affect the transferability of the input embeddings?

Context. After using a heuristic or numerical measure to assess the suitability of input embeddings for an NLP task, the question arises: what happens when the same method is employed now to evaluate the quality of the embeddings alongside the predicted labels generated by the model?

Qualitative investigation. This research question explores the relationship between the architecture of an NLP task, the linguistic knowledge encoded in pre-trained input embeddings, and their transferability. An extrinsic evaluation approach is employed to investigate this, where a model is trained using different representations. The objective is to determine whether models with high evaluation metrics also exhibit high dependency values. In this context, the feature selection measure is implemented on the test set, considering both the actual labels and the predicted ones produced by the model.

Statements. Additionally, since different embeddings ends-up producing similar results for the same model [Muennighoff et al. 2022], the following scenarios are also considered: (i) if the feature selection measure indicates high dependency values, yet the model still performs poorly, the issue may lie with the remaining network components of the model; and (ii), likewise, if the measure suggests low dependency values, but the model achieves high-quality results, subsequent layers beyond the textual representation may impact transferability. The ultimate goal of this research question is to determine the interchangeability of the mentioned statements.

4. Experimental Setup

4.1. Probing Task

The concept of probing tasks introduced by [Shi et al. 2016] and [Adi et al. 2016] involves using a pre-trained encoder (e.g., embeddings) to train a classifier or decoder that focuses on simple linguistic properties of sentences [Conneau et al. 2018]. If the classifier succeeds, the pre-trained encoder representations contain sufficient information to solve the task effectively. Given the distinct aspects of the research questions, the experimentation focused only on the Text Classification task as a probing task, specifically Sentiment Analysis[2].

[1] This work focuses on pre-trained and fine-tuned embeddings, which have become a trend in NLP systems and a key component of state-of-the-art models [Liu et al. 2019a].

[2] It is worth noting that this task belongs to the group of tasks used as an extrinsic evaluation method.

Table 1 summarizes the main statistics of the datasets used in this paper: one of them, CoLA [Warstadt et al. 2019], is part of the benchmark GLUE [Wang et al. 2018a], and the remaining three, IMDb [Maas et al. 2011], SST-2 [Socher et al. 2013], and Sentiment140 [Go et al. 2009], are generic datasets widely utilized for Text Classification/Sentiment Analysis task.

Sentiment Classification Dataset	# Corpus	# Class	Is Balanced?	# Per Class		Language
				0	1	
IMDb [Maas et al. 2011]	50,000	2	Yes	25,000	25,000	English
SST-2 [Socher et al. 2013]	68,219	2	No	30,207	38,012	
CoLA [Warstadt et al. 2019]	9,594	2	No	2,850	6,744	
Sentiment140 [Go et al. 2009]	160,000	2	Yes	79,849	80,151	

Table 1. Statistics of corpora used in experiments. To ensure comparability with other datasets, 10% of the Sentiment140 training data was randomly selected to maintain comparable text amounts. The classification values in the table correspond to combined subsets within each corpus, including training, test, or validation sets.

All corpora are in English and were loaded via Hugging Face[3]. Although Sentiment140 originally had three classes, only the available training set with two classes was used. Also, hold-out validation was conducted by combining all properly annotated texts from each corpus. The data was split into 70% training and 30% test sets.

4.2. Feature Selection Measure

Mutual Information (MI) is a statistical measure that quantifies the mutual dependence or information shared between two variables [Fano 1961]. In the context of NLP and embeddings, MI can be used to assess how well the embedding representation captures relevant information about the input text and its corresponding labels in a given task. However, there are some considerations to keep in mind:

Advantages. MI captures relevant information since it measures the relevance of the embedding representation to the task (i.e., higher MI indicates more relevant information). Additionally, MI can deal with non-linear dependencies since it helps model complex relationships between text and task targets.

Challenges. The accurate estimation of MI can be difficult, especially for high-dimensional embeddings. While MI provides a quantitative measure, understanding the specific linguistic or semantic aspects captured or neglected may be challenging.

In summary, MI can be a valuable tool for assessing the quality of embedding representations for NLP tasks [Zhelezniak et al. 2020]. To tackle the listed challenges, it is crucial to complement MI with other evaluation techniques. Incorporating task performance (i.e., qualitative analysis) will be essential in the results section.

Here, MI is used with Scikit-learn[4]. The `discrete_features` parameter was modified to consider continuous features. As mentioned as one of the challenges, using MI with dense representations has been difficult since it can have some issues estimating

[3] https://huggingface.co/datasets

[4] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html

MI for continuous random variables [Zhelezniak et al. 2020]. Yet, the aim is to assess Scikit's usability for such representations.

4.3. Probing Model and Parameters

A basic model architecture for Sentiment Analysis comprises two intermediate layers with 32 units each and a final layer for sentiment prediction with sigmoid activation. The models are trained using the Keras library, a high-level API of TensorFlow, for up to 300 epochs with early stopping and a patience of five.

4.4. Pre-Trained Embedding Models

The experimentation involves three widely used publicly available models for English embeddings: GloVe [Pennington et al. 2014] and fastText [Bojanowski et al. 2016], which are static/classic embeddings, and DistilBERT [Sanh et al. 2019], a contextual representation model that has 40% fewer parameters than the original BERT Base.

The pre-trained static embeddings GloVe[5] and fastText[6] were loaded via the Flair library for NLP[7], having 300-dimensional vectors each. The DistilBERT pre-trained model[8] was instantiated using the Transformers library from Hugging Face. By default, the hidden states of all Transformer-based model layers are concatenated to produce the embeddings, generating vectors with 768 dimensions.

Sentence embeddings. To accommodate the requirements of the Scikit-learn MI function, the pre-trained representations were employed as sentence embeddings. The pooling operation used for static and contextual embeddings gives the mean of all words in the sentence. The texts with the pre-trained static sentence embeddings were embedded via Flair. The sentence embedding matrices were extracted after training the models to be used as input to the MI function.

5. Experimental Results

5.1. RQ1 Results

Table 2 contains the performance of the models during training via accuracy results and some info about MI scores (maximum and mean values). Table 3 retains precise info about the MI scores distribution, showing descriptive statistics of the percentiles. The following observations can be made in greater detail:

Pre-trained sentence embedding representation	IMDb			SST-2			CoLA			Sentiment140		
	Training Acc	MI Scores		Training Acc	MI Scores		Training Acc	MI Scores		Training Acc	MI Scores	
	Best Epoch	Max	Mean	Best Epoch	Max	Mean	Best Epoch	Max	Mean	Best Epoch	Max	Mean
GloVe (300d)	0.879	0.052	0.011	0.950	0.067	0.037	0.923	0.012	0.012	0.792	0.032	0.011
fastText (300d)	0.881	0.070	0.010	0.963	0.065	0.035	0.874	0.014	0.002	0.817	0.040	0.011
Fine-tuned GloVe (300d)	0.985	0.556	0.306	**0.984**	0.279	0.097	**0.958**	0.060	0.014	**0.953**	0.155	0.027
Fine-tuned fastText (300d)	**0.987**	0.602	0.384	0.980	0.300	0.117	0.940	0.143	0.033	0.929	0.220	0.050
DistilBERT (768d)	0.908	0.089	0.013	0.918	0.117	0.040	0.911	0.022	0.003	0.794	0.040	0.008

Table 2. Model performance results during training and MI scores key information. The best results for the accuracy metric are highlighted in bold.

[5] https://nlp.stanford.edu/projects/glove

[6] https://fasttext.cc/docs/en/english-vectors.html

[7] https://flairnlp.github.io/docs/intro

[8] https://huggingface.co/docs/transformers/model_doc/distilbert

Pre-trained sentence embedding representation	IMDb			SST-2			CoLA			Sentiment140		
	MI Scores Percentiles											
	25th	50th	75th	25th	50th	75th	25th	50th	75th	25th	50th	75th
GloVe (300d)	0.005	0.008	0.015	0.329	0.036	0.040	0.000	0.000	0.003	0.009	0.010	0.013
fastText (300d)	0.004	0.007	0.012	0.030	0.033	0.037	0.000	0.000	0.003	0.008	0.009	0.013
Fine-tuned GloVe (300d)	0.207	0.323	0.434	0.049	0.074	0.128	0.003	0.012	0.022	0.010	0.019	0.036
Fine-tuned fastText (300d)	0.282	0.462	0.552	0.062	0.098	0.171	0.011	0.025	0.051	0.016	0.036	0.069
DistilBERT (768d)	0.005	0.009	0.018	0.036	0.037	0.044	0.000	0.001	0.005	0.004	0.006	0.010

Table 3. Descriptive statistics with the percentiles of the MI scores distribution for the sentence embeddings of each training set.

Fine-tuning process. The static sentence embeddings, such as those from GloVe and fastText, were fine-tuned during training. The results include the frozen and fine-tuned versions. Although this deviates from the standard approach of maintaining encoder architecture agnosticism in probing tasks, it allows an understanding of the extent of the dependency introduced by fine-tuning these static vectors. On the other hand, the fine-tuning of DistilBERT was not performed to enable a more focused investigation of its original internal layers and their contributions to the task.

High MI score values for fine-tuned representations. Indeed, as reported in Tables 2 and 3, fastText's updated embeddings showed the highest dependency across all datasets, closely followed by fine-tuned GloVe embeddings. On the other hand, the non-updated sentence embeddings generally had MI scores much closer to 0. Although DistilBERT has many dimensions with score values close to 0, it exhibits a more comprehensive range of scores among the non-tuned representations.

Good performance of probing models during training. Another observation is that most models converged well during training, obtaining accuracies close to or greater than 0.90, except for GloVe and DistilBERT on the Sentiment140 training data. The **RQ2** will verify if the models are ideal and are on the borderline between underfitting and overfitting.

RQ1 answer. As a response to **RQ1**, the MI measure does not readily indicates which sentence representations are sufficient to solve the different Sentiment Analysis tasks. Despite the MI measure lacking evidence, these results are still valuable as part of the research, which includes attempts beyond initial expectations.

Updating sentence embeddings during training logically improves the correlation between input embeddings and the task objective. Also, note that the MI values are low for CoLA and Sentiment140. This will be discussed further in the next section.

5.2. RQ2 Results

The objective of the **RQ2** is to observe the semantic transferability of embeddings with the MI measure and an extrinsic evaluation. The analysis conducted using the MI measure on the training set was similarly applied to the test set. Minor changes were expected in the distribution of MI scores, as the main characteristics of the data were retained for both sets. Tables 4 and 5 present descriptive statistics of the percentile distribution of MI scores, including actual and predicted classes. Further examination reveals the following detailed observations:

Conflicting MI score values for CoLA. As expected, the distributions are equivalent to

Pre-trained sentence embedding representation	**IMDb**			**SST-2**			**CoLA**			**Sentiment140**		
	MI Scores Percentiles											
	25th	**50th**	**75th**	**25th**	**50th**	**75th**	**25th**	**50th**	**75th**	**25th**	**50th**	**75th**
GloVe (300d)	0.003	0.007	0.014	0.014	0.017	0.022	0.000	0.001	0.006	0.007	0.010	0.011
fastText (300d)	0.002	0.006	0.011	0.015	0.018	0.021	0.000	0.001	0.006	0.007	0.010	0.013
Fine-tuned GloVe (300d)	0.168	0.264	0.336	0.029	0.053	0.102	0.000	0.002	0.008	0.011	0.018	0.032
Fine-tuned fastText (300d)	0.225	0.349	0.403	0.038	0.069	0.132	0.000	0.002	0.007	0.016	0.032	0.061
DistilBERT (768d)	0.003	0.008	0.017	0.016	0.019	0.026	0.000	0.002	0.007	0.003	0.005	0.009

Table 4. Descriptive statistics with the percentiles of the MI scores distribution for the sentence embeddings of each test set with the *actual* classes.

Pre-trained sentence embedding representation	**IMDb**			**SST-2**			**CoLA**			**Sentiment140**		
	MI Scores Percentiles											
	25th	**50th**	**75th**	**25th**	**50th**	**75th**	**25th**	**50th**	**75th**	**25th**	**50th**	**75th**
GloVe (300d)	0.003	0.008	0.015	0.016	0.019	0.023	0.003	0.008	0.014	0.012	0.014	0.017
fastText (300d)	0.003	0.008	0.014	0.015	0.018	0.023	0.005	0.011	0.017	0.012	0.015	0.022
Fine-tuned GloVe (300d)	0.184	0.283	0.384	0.031	0.056	0.106	0.006	0.014	0.022	0.015	0.025	0.039
Fine-tuned fastText (300d)	0.244	0.416	0.510	0.041	0.075	0.143	0.010	0.021	0.032	0.021	0.040	0.071
DistilBERT (768d)	0.003	0.009	0.020	0.016	0.020	0.027	0.000	0.003	0.009	0.004	0.007	0.014

Table 5. Descriptive statistics with the percentiles of the MI scores distribution for the sentence embeddings of each test set with the *predicted* classes.

those for **RQ1**. However, the CoLA dataset has the most incompatible distributions, with score values lower than those obtained for the training set. Note that for the predicted classes, CoLA's MI scores are much more similar to those for the training set, indicating that the model can accurately maintain the variance and patterns learned during training.

MI score values indicating overfitting for CoLA and Sentiment140. Also, for the CoLA dataset, it is possible to deduce, based on the MI scores, that the model may not have performed well for the test set, which indicates the existence of overfitting. Overfitting can also be noticed for Sentiment140 by some subtle changes when comparing the distributions of the scores of the predicted classes with the actual ones. The existence or not of overfitting in the models will be further analyzed. The fact that the MI scores could reveal this unexpected model behavior is intriguing.

Extrinsic evaluation. Table 6 reports the accuracies obtained between the ground truth and what was predicted by the probing models, and some info about the MI scores. Table 7 has more extrinsic evaluation results for other metrics.

Pre-trained sentence embedding representation	**IMDb**			**SST-2**			**CoLA**			**Sentiment140**		
	Test Acc	**MI Scores**		**Test Acc**	**MI Scores**		**Test Acc**	**MI Scores**		**Test Acc**	**MI Scores**	
		Max	**Mean**		**Max**	**Mean**		**Max**	**Mean**		**Max**	**Mean**
GloVe (300d)	0.830	0.050	0.010	0.879	0.049	0.019	0.625	0.024	0.004	0.500	0.267	0.010
fastText (300d)	0.857	0.072	0.009	0.887	0.051	0.019	0.609	0.021	0.004	0.500	0.042	0.010
Fine-tuned GloVe (300d)	0.879	0.406	0.241	0.910	0.242	0.072	0.577	0.023	0.004	0.497	0.139	0.025
Fine-tuned fastText (300d)	**0.880**	0.430	0.288	**0.912**	0.253	0.088	0.615	0.024	0.004	0.499	0.190	0.044
DistilBERT (768d)	0.866	0.097	0.013	0.867	0.094	0.022	**0.677**	0.027	0.004	**0.501**	0.044	0.007

Table 6. Model performance results for the test set and MI scores key information. The best results for the accuracy evaluating metric are highlighted in bold.

Overfitting for CoLA and Sentiment140. The models with the different sentence embeddings have high and very close accuracies for the IMDb and SST-2 datasets. It is possible to state through these results that the CoLA and Sentiment140 models suffered overfitting.

Pre-trained sentence	**IMDb**			**SST-2**			**CoLA**			**Sentiment140**		
embedding representation	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**	**P**	**R**	**F_1**
GloVe (300d)	0.832	0.831	0.831	0.881	0.879	0.879	0.627	0.626	0.627	0.500	0.500	**0.500**
fastText (300d)	0.857	0.857	0.857	0.887	0.887	0.886	0.618	0.609	0.613	0.500	0.500	0.499
Fine-tuned GloVe (300d)	0.880	0.880	**0.880**	0.911	0.910	0.910	0.623	0.577	0.594	0.497	0.497	0.496
Fine-tuned fastText (300d)	0.881	0.880	**0.880**	0.912	0.912	**0.912**	0.613	0.615	0.614	0.499	0.499	0.495
DistilBERT (768d)	0.868	0.866	0.866	0.867	0.867	0.866	0.703	0.677	**0.687**	0.500	0.501	0.499

Table 7. Extrinsic evaluation of sentence embeddings for different datasets. The best results for the weighted F_1-score (F_1) metric are highlighted in bold.

Extrinsic evaluation analysis. Looking only at the accuracy metric, it is undefined which sentence embedding is the best for a specific dataset, which usually happens for this type of embedding evaluation. As reported by accuracy, fastText and GloVe achieved the best F_1-score results among the sentence embeddings for the IMDb and GloVe corpora. DistilBERT was the best only for the CoLA dataset. Finally, it is imprecise, which is better for Sentiment140, neither by accuracy nor by the F_1-score.

RQ2 answer. Based on the results, the probing model designed for solving various Sentiment Analysis tasks impacts the input representations. As anticipated, the extrinsic evaluation results for different sentence embeddings exhibit high similarity, making it challenging to determine the best embedding model for particular datasets, such as IMDb and Sentiment140. It is worth mentioning that the low distribution of MI scores of all subsets for the CoLA and Sentiment140 corpora, even with fine-tuned sentence embeddings, suggests that the models would perform poorly for these datasets since the beginning.

Closing to the RQ2 statements. Only the second statement can be attested to the two expected scenarios: the model performs well even when the MI measure indicates low dependency values. This suggests that the subsequent layers in the model may impact semantic transferability. It is worth noting that the dimensions with high dependency MI score values are primarily associated with embedding models that underwent fine-tuning during training. However, fine-tuning can not be considered the optimal solution.

6. Conclusion

This study aimed to answer two research questions related to the suitability and transferability of input embeddings for NLP tasks. The first one, **RQ1**, focused on determining the dependence between different input embeddings on the target of the Sentiment Analysis task. The **RQ2** examined the transferability of the embeddings and their impact on the model's performance. Overall, the findings emphasize the importance of model architecture and highlight the complexities involved in evaluating the suitability and transferability of input embeddings for NLP tasks.

Research future directions. The results obtained for the research questions could have been better; therefore, future work should focus on improving results by expanding the experimental setup to include more NLP tasks for evaluation, researching feature selection measures for dense data, developing customized probing models, and comparing static vs. contextual embeddings. Additionally, consider including non-English languages for better generalizability. The main objective of these future works is to create a framework that addresses the need for adequate and high-level evaluation of different NLP systems, providing accurate and initial indications for their construction.

References

[Adi et al. 2016] Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., and Goldberg, Y. (2016). Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. *arXiv preprint arXiv:1608.04207*.

[Bakarov 2018] Bakarov, A. (2018). A Survey of Word Embeddings Evaluation Methods. *arXiv preprint arXiv:1801.09536*.

[Boggust et al. 2022] Boggust, A., Carter, B., and Satyanarayan, A. (2022). Embedding Comparator: Visualizing Differences in Global Structure and Local Neighborhoods via Small Multiples. In *27th International Conference on Intelligent User Interfaces*, pages 746–766.

[Bojanowski et al. 2016] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching Word Vectors with Subword Information. *arXiv preprint arXiv:1607.04606*.

[Butcher and Smith 2020] Butcher, B. and Smith, B. J. (2020). Feature Engineering and Selection: A Practical Approach for Predictive Models: by Max Kuhn and Kjell Johnson. Boca Raton, FL: Chapman & Hall/CRC Press, 2019, xv+ 297 pp., $79.95 (H), ISBN: 978-1-13-807922-9.

[Carter et al. 2019] Carter, B., Mueller, J., Jain, S., and Gifford, D. (2019). What made you do this? Understanding black-box decisions with sufficient input subsets. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 567–576. PMLR.

[Chen et al. 2018] Chen, J., Tao, Y., and Lin, H. (2018). Visual Exploration and Comparison of Word Embeddings. *Journal of Visual Languages & Computing*, 48:178–186.

[Conneau et al. 2018] Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. (2018). What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*.

[Fano 1961] Fano, R. M. (1961). Transmission of Information: A Statistical Theory of Communications. *American Journal of Physics*, 29(11):793–794.

[Go et al. 2009] Go, A., Bhayani, R., and Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. *CS224N project report, Stanford*, 1(12):2009.

[Hamilton et al. 2016] Hamilton, W. L., Leskovec, J., and Jurafsky, D. (2016). Cultural Shift or Linguistic Drift? Comparing Two Computational Measures of Semantic Change. In *Proceedings of the conference on empirical methods in natural language processing. Conference on empirical methods in natural language processing*, volume 2016, page 2116. NIH Public Access.

[Heimerl and Gleicher 2018] Heimerl, F. and Gleicher, M. (2018). Interactive Analysis of Word Vector Embeddings. In *Computer Graphics Forum*, volume 37, pages 253–265. Wiley Online Library.

[Ignat et al. 2023] Ignat, O., Jin, Z., Abzaliev, A., Biester, L., Castro, S., Deng, N., Gao, X., Gunal, A., He, J., Kazemi, A., et al. (2023). A PhD Student's Perspective on Research in NLP in the Era of Very Large Language Models. *arXiv preprint arXiv:2305.12544*.

[Jurafsky and Martin 2018] Jurafsky, D. and Martin, J. H. (2018). Speech and Language Processing. *preparation [cited 2020 June 1] Available from: https://web. stanford. edu/˜ jurafsky/slp3*.

[Li et al. 2018] Li, Q., Njotoprawiro, K. S., Haleem, H., Chen, Q., Yi, C., and Ma, X. (2018). EmbeddingVis: A Visual Analytics Approach to Comparative Network Embedding Inspection. In *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 48–59. IEEE.

[Liu et al. 2019a] Liu, N. F., Gardner, M., Belinkov, Y., Peters, M. E., and Smith, N. A. (2019a). Linguistic Knowledge and Transferability of Contextual Representations. *arXiv preprint arXiv:1903.08855*.

[Liu et al. 2019b] Liu, Y., Jun, E., Li, Q., and Heer, J. (2019b). Latent Space Cartography: Visual Analysis of Vector Space Embeddings. In *Computer graphics forum*, volume 38, pages 67–78. Wiley Online Library.

[Maas et al. 2011] Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

[Muennighoff et al. 2022] Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. (2022). MTEB: Massive Text Embedding Benchmark. *arXiv preprint arXiv:2210.07316*.

[Oliveira et al. 2022] Oliveira, B. S. N., do Rêgo, L. G. C., Peres, L., da Silva, T. L. C., and de Macêdo, J. A. F. (2022). Processamento de Linguagem Natural via Aprendizagem Profunda. *Sociedade Brasileira de Computação*.

[Pennington et al. 2014] Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

[Ribeiro et al. 2016] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why Should I Trust You?" Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

[Sanh et al. 2019] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

[Schnabel et al. 2015] Schnabel, T., Labutov, I., Mimno, D., and Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 298–307.

[Shi et al. 2016] Shi, X., Padhi, I., and Knight, K. (2016). Does String-Based Neural MT Learn Source Syntax? In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1526–1534.

[Shrikumar et al. 2017] Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning Important Features Through Propagating Activation Differences. In *International conference on machine learning*, pages 3145–3153. PMLR.

[Socher et al. 2013] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

[Torregrossa et al. 2021] Torregrossa, F., Allesiardo, R., Claveau, V., Kooli, N., and Gravier, G. (2021). A survey on training and evaluation of word embeddings. *International Journal of Data Science and Analytics*, 11(2):85–103.

[Wang et al. 2018a] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018a). Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

[Wang et al. 2018b] Wang, Y., Liu, S., Afzal, N., Rastegar-Mojarad, M., Wang, L., Shen, F., Kingsbury, P., and Liu, H. (2018b). A Comparison of Word Embeddings for the Biomedical Natural Language Processing. *Journal of biomedical informatics*, 87:12–20.

[Warstadt et al. 2019] Warstadt, A., Singh, A., and Bowman, S. R. (2019). Neural Network Acceptability Judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

[Zhelezniak et al. 2020] Zhelezniak, V., Savkov, A., and Hammerla, N. (2020). Estimating Mutual Information Between Dense Word Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8361–8371.