Understanding the learning dynamics of word2vec

Stanford CS224N Custom Project

Alvin Tan
Symbolic Systems Program
Stanford University
tanawm@stanford.edu
Mentor: Chris Manning

Abstract

Word embeddings are a common representation of linguistic data used in a variety of natural language processing (NLP) applications. However, it is not exactly clear what kind of information is being learnt during the training, and why it results in embeddings that can serve as effective inputs for a variety of downstream NLP tasks. This project studied how a variety of evaluation metrics change dynamically over the course of training in word2vec, including both intrinsic evaluators (word similarity and word analogy) and extrinsic evaluators (part-of-speech tagging and named entity recognition), as well as a set of linguistics diagnostics. We found that the metrics exhibited a number of clusters, including one cluster containing word similarity, part-of-speech tagging, and named entity recognition, which are tasks related to word categorisation. Another cluster contained word analogy, as well as proportion of neighbours which are synonyms or associations, which are tasks that rely more finely on the particular semantics of words. Together, these suggest that word2vec learns how to categorise words with similar properties, before fine-tuning the specific embeddings of words.

1 Introduction

Word embeddings are a common representation of linguistic data, used in a variety of natural language processing (NLP) applications. One common word embedding model is the Skip-gram model of word2vec [1], which aims to predict context words within a window given a central word. This algorithm is straightforward to understand statistically, and a number of subsequent analyses have further provided theoretical formulations for word2vec (e.g. [2, 3, 4]).

However, the interpretability of this algorithm remains challenging. It is not clear how simple cooccurrence statistics in the input can result in models that capture meaningful linguistic information, including morphosyntactic and semantic information, which enables such models to perform well in various downstream NLP tasks. [5] In particular, there is very little work studying the learning dynamics of word2vec. Typically, the only time-course information presented about such models reflect scores related to the objective function (e.g. loss, accuracy, F-score); however, this does not reflect the kinds of information learnt by the model, and how they vary over time.

As such, this project aims to understand how word2vec models change over the course of training. We take snapshots of the model during training, and run evaluations and diagnostics on these snapshots to understand how the linguistic information captured by the model evolves over time. The results from these analyses thus help to render the learning process of word2vec more interpretable.

2 Related Work

2.1 Interpretability

For machine learning applications, interpretability is important as it allows discovery, explanation, control, and improvement of models [6]. Work in interpetability can generally be classified into two categories [7]. The first is model-based interpretability, which refers to the design of models with greater interpretability, either through having a simpler design, or by including proxies to interpretability in the objective function. In particular, some research in word embedding models have achieved this through explicitly encouraging words related to a particular concept to take large values along a corresponding dimension [8], or through optimising for sparsity [9], among other approaches.

The second approach is post-hoc interpretability, which refers to interpreting a model after it has been trained through various types of probes. For example, it is possible to apply rotations [10] or projections [11] to trained embedding models to improve their interpretability. Rogers et al. [12] also proposed the Linguistic Diagnostics method, which measures properties of the embedding space directly by quantifying relations that hold within neighbourhoods of a sample of words.

2.2 Evaluation

Simultaneously, models must be performant enough to be useful for application. There have been many proposed methods of evaluating the performance of word embedding models (see e.g. [13, 14]), which broadly fall into the categories of intrinsic and extrinsic evaluators. The former refer to evaluators which examine the embeddings themselves in comparison with human judgements on words, while the latter refer to evaluators which use embeddings for downstream NLP tasks. In particular, Wang et al. [14] demonstrated that evaluators do not perform in parallel across models. As such, it stands to reason that the properties captured by such evaluators may be learnt at different rates during training.

2.3 Dynamics

There is much less work studying the dynamics of machine learning model. One recent example is a work by Chang and Bergen [15] studying word acquisition in language models, quantified by change in surprisal for a word over training. This approach allowed a greater understanding of the learning process of the model, as well as the factors influencing its learning.

Drawing these three branches of related work together, the present project aims to evaluate a word embedding model across time, permitting interpretation of the learning dynamics of the model and its acquisition of linguistic information.

3 Approach

3.1 Training

We trained a word2vec model with Skip-gram negative sampling using the Gensim [16] port of the original toolkit, with the default settings.[1] We used the wiki2010 corpus [17] as the training data, and took snapshots of the models every 100M words (i.e. 10 snapshots per epoch $\times$ 5 epochs = 50 snapshots per model). The randomly initialised model (i.e. the 0^{th} snapshot) was also used as an (internal) baseline.

3.2 Evaluation and diagnostics

These snapshots were evaluated on a set of intrinsic and extrinsic evaluators, which are a subset of tasks from Wang et al. [14]. We selected tasks which would capture a variety of different aspects of linguistic information (e.g. morphosyntactic and semantic), while reducing the number of tasks to

[1] Window size = 5, negative samples = 5, negative sampling exponent = 0.75, learning rate = linear decrease over [0.025, 0.0001]

ensure that computation is manageable (given that evaluation occurred over a much larger number of embedding models).

We also measured the embedding models using the Linguistic Diagnostics Toolkit [12], which provides metrics for a number of more granular linguistic properties. In particular, these diagnostics reflect the relationships between words and their k closest neighbours, which provides a representation of the organisation of the embedding space.

3.3 Analysis

The resultant evaluation and diagnostic metrics were collated and visualised to understand their dynamics across the training of word2vec. We then conducted a correlation analysis to understand how the various metrics clustered together.

4 Experiments

4.1 Evaluation

We employed two intrinsic evaluators—word similarity and word analogy—as well as two extrinsic evaluators—part-of-speech (POS) tagging and named entity recognition (NER). The list of evaluation tasks, as well as their corresponding datasets and evaluation metrics, is displayed in Table 1.

Table 1: List of intrinsic and extrinsic evaluators employed.

Task	**Dataset**	**Metric**
Word similarity	WordSim-353 [18]	Correlation (using cosine similarity)
Word analogy	BATS [19]	Accuracy (using LRCos [20])
POS tagging	PTB [21]	Accuracy
NER	CoNLL'03 [22]	F-score
SA	IMDb [23]	Accuracy

For word similarity, we used the inbuilt evaluation function in Gensim [16], which measures the cosine similarity between words and the Pearson's correlation between the similarity value and human-rated similarities. We measured this on WordSim-353-Rel and WordSim-353-Sim [18], which are two subsets of WordSim-353 [24] that specifically include related words and similar words respectively.

For word analogy, we used the inbuilt evaluation function in Vecto [20], using the state-of-the-art LRCos method [20] to measure performance on four balanced subsets of BATS [19]: inflectional morphology, derivational morphology, encyclopaedic semantics, and lexicographic semantics.

For POS tagging and NER, we used a window-based feed-forward neural network with a window size of 5, with inputs fed into a 300-unit hidden layer, followed by a hard tanh activation, then a fully-connected output layer. Each model was trained for 10 epochs using the Adam optimiser [25] with a batch size of 50 and a learning rate of 0.001. As in Wang et al. [14], the code for these tasks were adapted from Chiu et al. [26], updated to conform to Python 3.

For SA, we used a convolutional neural network with filter sizes [3, 4, 5] and 100 filters per size. The convolved outputs were passed through a ReLU activation, then max-pooled, concatenated, and passed through a dropout layer, then a fully-connected output layer. Each model was trained for 5 epochs using the Adam optimiser with a batch size of 50 and a learning rate of 0.0001. The code for this task was adapted from [27], which is a reimplementation of the algorithm from Kim [28].

For all the extrinsic evaluators, we used the standard splitting ratios for the train, validation, and test sets. All embeddings were frozen during extrinsic evaluation.

4.2 Diagnostics

We used the Linguistic Diagnostics Toolkit [12] to measure properties of the embedding spaces. This process involves choosing a sample of words, finding their k closest neighbours, and measuring the relationships between the sampled word and its neighbourhood.

We subsampled the original ldt909 wordlist to maintain a balanced sample across word frequency and part of speech, additionally controlling for polysemy. This resulted in a total sample of 95 words (ldt95).[2] The distribution of the sample is shown in Table 2

Table 2: Distribution of words in ldt95 (monosemous / polysemous).

Frequency bin	**Nouns**	**Verbs**	**Adjectives**	**Adverbs**
100 1,000	3/3	3/3	3/3	3/3
1,000 10,000	3/3	3/3	3/3	3/3
10,000 100,000	3/3	3/3	3/3	3/3
>100,000	3/3	2/3	3/3	3/3

We used $k = 10$ and measured neighbourhood properties in shared morphology (shared POS, shared morphological form, shared derivation), semantic relations (synonyms, antonyms, meronyms, hyponyms, hypernyms), and psychological relations (associations). The resultant diagnostics reflect the proportion of neighbours which satisfy each relationship with the sampled word.

4.3 Analysis

We calculated the Pearson's correlation coefficient between all pairs of metrics using the `FactoMineR` package in R, and plotted the values on a correlogram.

5 Results

5.1 Dynamics

The dynamics of the various evaluators and linguistic diagnostics are shown in Figure 1.

Qualitatively, there appear to be some metrics which grow rapidly initially, and then plateau relatively quickly (e.g. POS tagging, NER, word similarity). In contrast, there are other metrics which continue to increase over the course of training (e.g. BATS, synonyms, associations). This suggests that there are some properties of the embedding space which are quickly acquired, whereas others are learnt more gradually over the course of training.

5.2 Correlation analysis

The correlogram of the evaluators and linguistic diagnostics is shown in Figure 2.

There are three particular clusters of tasks that exhibit high correlation ($r > .8$ for most correlations within the cluster). The first includes both word similarity measures, as well as POS tagging and NER. Conceptually, these tasks rely on categorisation of words—grouping words with similar properties (e.g. sharing semantic features or POS), such that high performance is achievable when word categories occupy small-enough clusters within the embedding space.

The second cluster includes the analogy tasks, as well as the synonyms and associations diagnostics. Conceptually, these tasks rely more on the specific embeddings of words, such that the exact direction of the word vector is much more important. This would result in synonyms and associations occurring close together within small neighbourhoods, and would also allow for the elucidation of the correct result for analogy tasks (which rely on "parallel" differences between word vectors).

A third cluster involves the shared POS and shared morphological form diagnostics. This cluster seems to be less important as it does not correlate strongly with any evaluation task. However, it does suggest that morphological form is one aspect of words that embedding models learn quickly, perhaps due to the fact that words of a similar form (i.e. sharing affixes, or both being lemmas) often occur in similar subcategorisation frames.

[2]The incomplete highest frequency bin for verbs is due to discarding words belonging to multiple POS, as in the original ldt909.

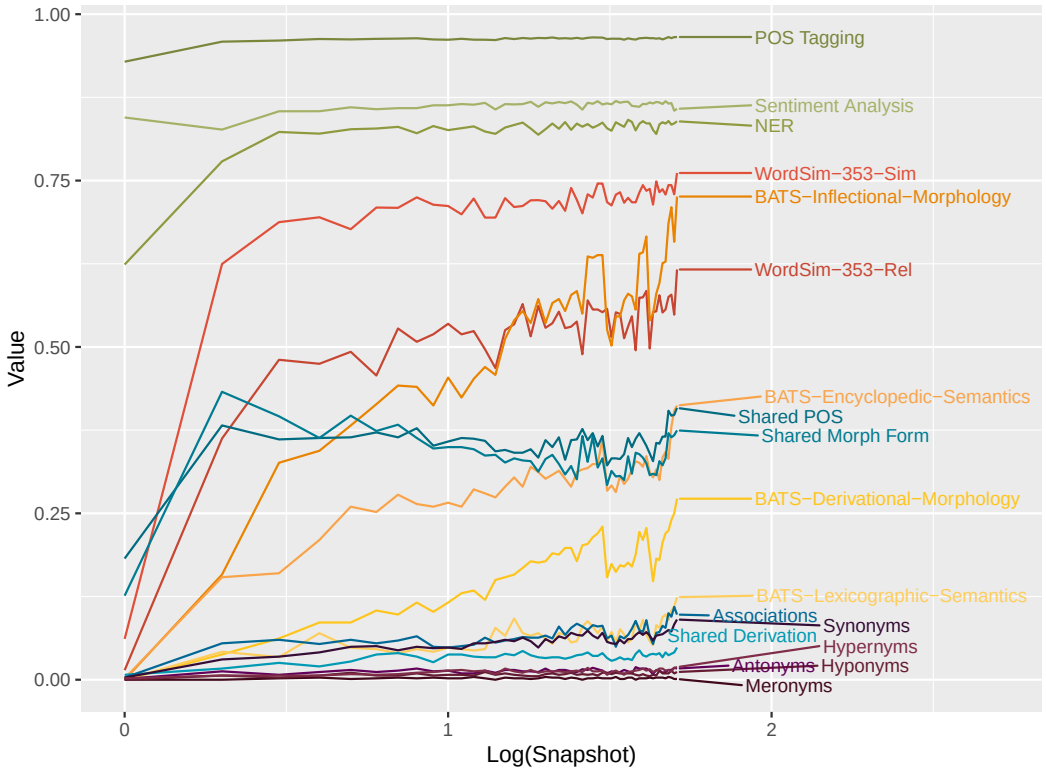

Figure 1: Dynamics of evaluation and diagnostic metrics over training.

6 Discussion

Over the course of training, word2vec first learns word similarity, POS tagging, and NER, followed by analogy, synonyms, and associations. This suggests that word2vec first learns to cluster words into relevant categories, which are then maintained as the specific semantics (i.e. specific directions of the embeddings) are fine-tuned. Notably, this occurs without any direct training on categorisation; rather, the explicit objective function merely relates to distributional characteristics of the training corpus. This suggests that the relevant categories that arise result directly from the cooccurrence statistics of the text; for example, nouns are commonly found before verbs in English, since subjects are often nouns.

These results also differ in interesting ways from previous results, including those from Rogers et al. [12] and Wang et al. [14]. Notably, they found lower, or even negative, correlations between word similarity and POS tagging/NER. The key difference between their studies and the present project is that they examined metrics across different *models*, whereas our project examined metrics across different *time points* of the same model. This suggests that the relationship among these evaluators holds specifically for word2vec, rather than being a relationship inherent in the task of learning word embeddings.

Indeed, more dynamics research is required to understand the learning processes of other word embedding models, and to determine which aspects are model-agnostic and which are model-dependent. For example, models that include subword representations (e.g. [29, 30, 31]) may learn morphology more quickly, since much of English inflectional and derivational morphology is marked by overt affixes that are much more easily captured at the subword level. In contrast, encyclopaedic information is likely to require substantial training for any model, since such information is learnable only via exposure to such information in the training corpus. We also hypothesise that the categorisation-then-tuning learning paradigm is likely to be consistent across models, since it seems to depend on statistics in the input rather than the specific optimisation function.

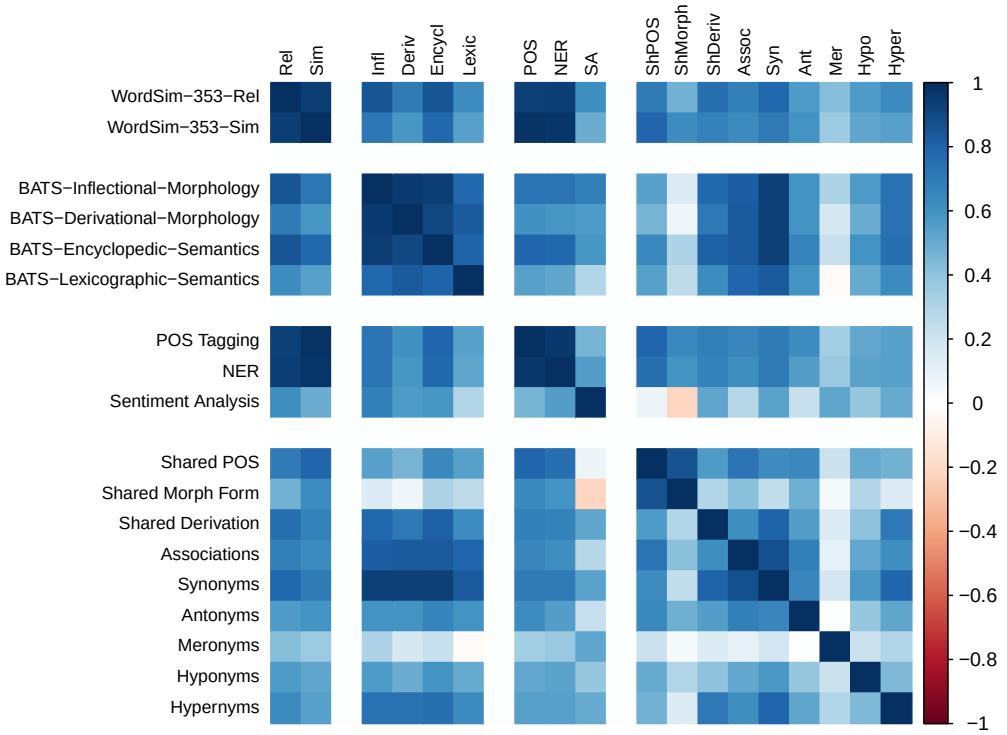

Figure 2: Correlogram of evaluation and diagnostic metrics.

Another point of interest is the relatively flat performance curves of the extrinsic evaluators. Notably, all three tasks reached within 2% of optimum performance within 4 snapshots (i.e. under 0.5 epochs). This suggests that the benefit of pretraining embeddings is quickly saturated, and subsequent pretraining does not significantly contribute to improved downstream performance. Since other tasks (e.g. analogy) do not show this pattern of early plateauing, it is unlikely that this is due to the embeddings remaining stagnant in quality. Rather, this suggests that these particular downstream tasks do not rely on very fine-grained information captured in word embeddings.

One important limitation to note that the concept of "categorisation" was inferred from the theoretical foundations of the evaluators used in this project, rather than measured directly. As such, there remains important future work that directly measures change in categorisation over time. One way to approach this relies on existing datasets of categories (in fact, possibly including the BATS datasets), and measuring the extent to which they form meaningful clusters (e.g. by measuring the average distance to the centroid across categories). This would allow us to explicitly test the hypothesis that word2vec learns about categorisation early and quickly.

7 Conclusion

In summary, word2vec learns to categorise words more quickly, then fine-tunes specific embeddings within those categories. As such, rather than acquiring specific dimensions of linguistic information at different rates (e.g. morphosyntactic vs semantic information), the structure of the embedding space changes at different rates over training. This result helps to render the training process of word2vec more interpretable, and further supports the utility of research studying the dynamics of machine learning models as a method of understanding their learning processes.

References

[1] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv e-prints*, page arXiv:1301.3781, January 2013.

[2] Omer Levy and Yoav Goldberg. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 171–180, June 2014.

[3] Yitan Li, Linli Xu, Fei Tian, Liang Jiang, Xiaowei Zhong, and Enhong Chen. Word embedding revisited: A new representation learning and explicit matrix factorization perspective. In *International Joint Conference on Artificial Intelligence*, 2015.

[4] Oren Melamud and Jacob Goldberger. Information-theory interpretation of the skip-gram negative-sampling objective function. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 167–171, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[5] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 01 2021.

[6] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160, 2018.

[7] W. James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019.

[8] Lütfi Kerem Şenel, İhsan Utlu, Furkan Şahinuç, Haldun M. Ozaktas, and Aykut Koç. Imparting interpretability to word embeddings while preserving semantic structure. *Natural Language Engineering*, 27(6):721–746, 2021.

[9] Abhishek Panigrahi, Harsha Vardhan Simhadri, and Chiranjib Bhattacharyya. Word2Sense: Sparse interpretable word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5692–5705, Florence, Italy, July 2019. Association for Computational Linguistics.

[10] Sungjoon Park, JinYeong Bak, and Alice Oh. Rotated word vector representations and their interpretability. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 401–411, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[11] Binny Mathew, Sandipan Sikdar, Florian Lemmerich, and Markus Strohmaier. The POLAR framework: Polar opposites enable interpretability of pre-trained word embeddings. In *Proceedings of The Web Conference 2020*, pages 1548–1558, 2020.

[12] Anna Rogers, Shashwath Hosur Ananthakrishna, and Anna Rumshisky. What's in your embedding, and how it predicts task performance. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2690–2703, 2018.

[13] Amir Bakarov. A survey of word embeddings evaluation methods. *CoRR*, abs/1801.09536, 2018.

[14] Bin Wang, Angela Wang, Fenxiao Chen, Yuncheng Wang, and C.-C. Jay Kuo. Evaluating word embedding models: Methods and experimental results. In *APSIPA Transactions on Signal and Information Processing*, volume 8, page e19, 2019.

[15] Tyler A. Chang and Benjamin K. Bergen. Word acquisition in neural language models. *CoRR*, abs/2110.02406, 2021.

[16] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.

[17] Cyrus Shaoul and Chris F. Westbury. The Westbury Lab Wikipedia Corpus, 2010.

[18] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A study on similarity and relatedness using distributional and WordNet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, June 2009.

[19] Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15, June 2016.

[20] Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3519–3530, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[21] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[22] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.

[23] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 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, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[24] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. *ACM Trans. Inf. Syst.*, 20(1):116–131, 2002.

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[26] Billy Chiu, Anna Korhonen, and Sampo Pyysalo. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 1–6, Berlin, Germany, August 2016. Association for Computational Linguistics.

[27] Ben Trevett. Bentrevett/pytorch-sentiment-analysis: Tutorials on getting started with pytorch and torchtext for sentiment analysis., Dec 2017.

[28] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[29] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. Enriching word vectors with subword information. *CoRR*, abs/1607.04606, 2016.

[30] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomás Mikolov. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759, 2016.

[31] Phong Ha, Shanshan Zhang, Nemanja Djuric, and Slobodan Vucetic. Improving word embeddings through iterative refinement of word- and character-level models. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1204–1213, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics.