Fine-tuning Distributional Semantic Models for Closely-Related Languages

Kushagra Bhatia
Macquarie Group Limited,
Gurugram
kushagra.bhatia@macquarie.com

Divyanshu Aggarwal
Dept of Biotechnology,
DTU Delhi
divyanshuggrwl@gmail.com

Ashwini Vaidya
Dept of Humanities and
Social Sciences, IIT Delhi
avaidya@hss.iitd.ac.in

Abstract

In this paper we compare the performance of three models: SGNS (skip-gram negative sampling) and augmented versions of SVD (singular value decomposition) and PPMI (Positive Pointwise Mutual Information) on a word similarity task. We particularly focus on the role of hyperparameter tuning for Hindi based on recommendations made in previous work (on English). Our results show that there are language specific preferences for these hyperparameters. We extend the best settings for Hindi to a set of related languages: Punjabi, Gujarati and Marathi with favourable results. We also find that a suitably tuned SVD model outperforms SGNS for most of our languages and is also more robust in a low-resource setting.

1 Introduction

The development of word embedding models in NLP has led to improved performance on a range of lexical semantic tasks (Baroni et al., 2014). The SGNS (skip-gram negative sampling) training method has been shown to outperform previously used count-based models such as PPMI (Positive Point based Mutual Information) and SVD (truncated Singular Value Decomposition).

In this paper, we look at the task of word similarity in Hindi and related languages Punjabi, Gujarati and Marathi. Specifically, we experiment with hyperparameter tuning for SGNS, SVD and PPMI for Hindi and then ask whether the same hyperparameters can be extended and applied to typol related languages. We make use of the hyperparameters formulated in Levy et al. (2015) to tune all three models. We find that a suitably tuned SVD model *outperforms* SGNS. This result differs from Levy et al. (2015) which shows that fine-tuned SGNS and SVD models perform at par. We find that our Hindi SVD results are better than multilingual fastText (Grave et al., 2018) and the recently released IndicNLP Suite (Kakwani et al., 2020).

We hypothesize that hyperparameters are sensitive to linguistic properties. If this is true, then we should find similar results in languages that are typologically related to Hindi. Our results suggest that these hyperparameter settings for Hindi can be extended to typologically-related languages. Indeed, the results show that adapting the hyperparameters from Hindi is more advantageous as compared to the default settings or the settings recommended for English in Levy et al. (2015).

While reasonably large resources and datasets exist for the major Indian languages, Joshi et al. (2020) have shown that more than half of the Indo-Aryan languages represented in Wikipedia can be classified as having 'poor resource availability'. Given these limitations, a model that is able to generate representations that are robust in the face of less data can be advantageous. In our experiments, we take successively smaller corpus slices from our Hindi, Marathi, Gujarati and Punjabi corpora in order to test our models' performance. We find that SVD is more robust compared to other models in a low-resource setting. Sahlgren and Lenci (2016) have investigated the effects of data size on distributional semantic models and their results on the robustness of SVD correspond with our findings for Hindi and related languages.

In this paper, we first describe the creation of a new word similarity dataset for Hindi, which addresses some of the limitations of existing evaluation datasets. We then discuss the hyperparameter settings suggested in Levy et al. (2015) and describe our results for Hindi and related languages on the entire corpus as well as on smaller corpus sizes. We conclude with a summary of our findings.

Proceedings of the 8th VarDial Workshop on NLP for Similar Languages, Varieties and Dialects, pages 60–66
April 20, 2021 ©2021 Association for Computational Linguistics

2 Word Similarity Dataset

In order to evaluate our distributed semantic models, we carry out an intrinsic evaluation of the models based on word similarity. For the languages in our study, the currently available word similarity datasets include translated versions of English WordSim-353 (WS-353) (Akhtar et al., 2017).

However, we note that WordSim-353 (Finkelstein et al., 2001) has been criticized for conflating association and similarity in its word-pair annotation guidelines (Hill et al., 2015). As a consequence, WS-353 measures association rather than similarity. In addition to this, we observe that the Hindi version of WS-353 consists of numerous transliterations from English. In order to develop a more robust evaluation dataset for our word experiments, we created Hin-RG63, the Hindi version for English RG-65 (Rubenstein and Goodenough, 1965). This dataset carefully dissociates similarity and relatedness in its annotation guidelines and has been used as a benchmark for SemEval tasks (Camacho-Collados et al., 2015). While we were unable to create RG-65 translations for the other languages included in this paper, we plan to extend the work done for Hindi to more languages in the future.

Two native speakers of Hindi, who were also bilinguals provided the translation of words in English RG-65 to Hindi. A third translator moderated any disagreements. It is noteworthy that two of the word pairs from English RG-65 did not have any suitable distinct translation and hence were not included in the final dataset. 16 Hindi native speakers were presented with the similarity scoring guidelines given in Jurgens et al. (2014). The annotators were presented with a practice session consisting of sample word pairs before rating the actual Hindi word pairs in the dataset. Next, the annotators were asked to score each pair on a scale of 0 to 4. To present more flexibility, scoring with a step of 0.5 was permitted.

We computed inter-annotator agreement using pairwise correlation between individual annotators' ratings. Pearson and Spearman correlation coefficients are used to assess the linear correlation and monotonic relationship respectively. We report an average pairwise Pearson correlation of 0.814, and average pairwise Spearman correlation of 0.805 for Hin RG-63, our version of English RG-65. We make the dataset available for public use.[1]

3 Comparing SVD, PPMI and SGNS

Baroni et al. (2014)'s paper compared word embedding models or 'context-predicting models' like SGNS with 'context-counting' models like SVD on various lexical semantic benchmarks. Their results showed the superiority of context-predicting models. In a follow-up to this result, Levy et al. (2015) demonstrated that suitably augmented and tuned PMI (Pointwise Mutual Information) and SVD (Singular Value Decomposition) i.e context-counting models can perform at par with word embedding models. In fact, insights from word embeddings can be used to *augment* count-based models, resulting in only very small differences in performance.

These system design changes formulated as a set of transferable hyperparameters in Levy et al. (2015) were applied either at the pre-processing or post-processing stage, modifying the word vectors generated from these methods. The following section expands upon these hyperparameters

3.1 Hyperparameters

A major contribution of the Levy et al. (2015) study was the formulation of hyperparameters for context-counting models that are inspired by context predicting models. Such adaptations are feasible due to an overlap between the mathematical objectives of the two, which improve the performance of the traditional methods. The authors used a large English corpus with 1.5 billion tokens for their experiments and inferred that the differences between the two families of models are trivial.

All three methods, viz. SGNS, SVD and PPMI output the word vector representation. Following the same nomenclature for hyperparameters as Levy et al. (2015), we summarize the hyperparameters used in our experiments in Table 1.

For context-predicting models, cds is the smoothing factor to which the context count is raised in the unigram distribution for negative sampling. In Levy et al. (2015) cds has been adapted for PPMI and SVD. We examine values for cds= 0.75, otherwise the standard unigram sampling distribution is followed ($cds = 1$).

The hyperparameter neg denotes the number of negative samples for context-predicting models.

[1] https://github.com/ashwinivd/similarity_hindi

Hyperparameter	Abreviation	Search Space
Context Distribution Smoothing	cds	0.75, $1^\dagger$
Eigenvalue Weighting	eig	0, 0.5, $1^\dagger$
Shifted PPMI	neg	$1^\dagger$, 5, 15
Context Vector Addition	$w + c$	only $w^\dagger$, w+c
Window Size	win	$2^\dagger$, 3, 5

Table 1: Hyperparameter search space studied in our work.† Default settings. The hyperparameter $w + c$ is used for SVD and SGNS, not PPMI. eig is only used for SVD.

This translates to the amount that the PPMI matrix is shifted for context-counting (PPMI and SVD) models.

Eigenvalue weighting (eig) represents the exponent to the eigenvalue matrix in the word vector representation equation, obtained after factorization of the PPMI matrix. The values $eig = 0\ and\ 0.5$ lead to the symmetric versions of SVD, with the prior version completely removing the eigenvalue matrix from the representation.

Pennington et al. (2014) introduce the concept of context vector addition ($w + c$) to the word vector output by the model. Following the same idea, we check whether such an addition at post processing is beneficial for SGNS and SVD methods. The window size (win) is the range in which the context words are chosen on both sides of the analyzed word.

4 Experiments

4.1 Datasets

The model training was performed using publicly available monolingual corpora. For Hindi we used HindMonoCorp (Bojar et al., 2014) and for the other languages viz. Marathi, Gujarati and Punjabi, IndicCorp (Kakwani et al., 2020) is used. The text is pre-processed by removing punctuation, followed by normalization using the Indic NLP Library[2]. The statistics for each corpora is shown in Table 2 with the vocabulary size calculated after ignoring words appearing less than 100 times. We vary the size of corpus used for training the methods. For the experiments in a low-resource setting, corpus slices are created by randomly sampling a fraction of sentences from the entire corpus (4.3.2).

Language	Tokens	Sentences	Vocabulary
Hindi	786M	44M	100,667
Gujarati	719M	41.1M	158,445
Punjabi	773M	29.2M	82,512
Marathi	551M	34M	155,113

Table 2: Statistics of corpora used for model training. Values in million (M)

Evaluation of models trained on different languages is performed on WS235, annotated by Akhtar et al. (2017). We further evaluate the Hindi models on our very own Hin-RG65. The average Spearman correlation between the vector cosine similarity and the human rating of the word-pairs is used to rank the word representations.

4.2 Hyperparameter tuning

We study the impact of different hyperparameters on the performance, by evaluating the models trained on the complete HindMonoCorp with different configrations. A few pre-processing hyperparameters viz. deletion of rare words prior to creation of context window, dynamic context weighting and subsampling were only analyzed in the preliminary stage of experiments. These hyperparameters did not have much impact on the performance, and were not investigated further. We summarize the advantageous configurations of the hyperparameters shown in Table 1, along with the observed differences from Levy et al. (2015)'s recommendations for English.

Levy et al. (2015) advocated the use of $cds = 0.75$ for all 3 models: SGNS, PPMI and SVD. However, we do not observe a persistent trend for context distribution smoothing (cds). Although PPMI shows slight improvement with $cds = 0.75$, SVD and SGNS do not show any preferences.

For English it was observed that SVD performed better with a shorter window ($win = 2$), whereas SGNS did not show any preferences for win. In contrast, we observe a tendency of both the methods towards a larger window size ($win = 5$). Such a trend may be attributed to the difference in linguistic properties and morphology of the two languages. PPMI however performed best with $win = 2$.

Levy et al. (2015)'s results for English, show that a value of neg as 5 or 15 was equally beneficial for SGNS. For our work on Hindi, it showed a clear

[2] https://pypi.org/project/indic-nlp-library/

preference for $neg = 15$. Any value below this was not beneficial. We think this may be due to the relatively higher vocabulary-to-token ratio of the Hindi training corpus (as compared to English). Similarly, in the case of the context vector $w + c$, Levy et al. (2015) are equivocal about its impact, but we found that addition of a context vector $(w + c)$ always yielded an improvement in performance of SGNS.

In concordance with Levy et al. (2015), we observe substantial gains for SVD when the eigenvalue matrix is removed from the word vector equation (i.e. $eig = 0$), over $eig = 1$ (the default value) or 0.5.

After filtering out the top performing set of hyperparameters on the Hindi corpus, we validate whether our inferences hold on the other three inspected Indo-Aryan languages.

4.3 Training and Evaluation

For all four languages, we trained a 500-dimensional representation for all the models.

We investigate the performance gains when training with optimal configuration. With this aim, we evaluate our techniques trained on the complete corpus. Finally, in order to analyse the trends for low-resource scenarios, we report the performance of the fine-tuned models when trained with varying corpus sizes.

It should be noted that the hyperparameter search is independently carried out on the evaluation set for Hindi. This gives us an upper limit on the method's performance and highlights the importance of suitable hyperparameters. In a real setting however, we would require a dedicated development set for tuning the models. For Gujarati, Marathi and Punjabi, we are not tuning the hyperparameters over the evaluation set and are simply adapting to the recommended configurations obtained from Hindi.

Language	**PPMI**	**SVD**	**SGNS**
WS-235			
Hindi	0.541	0.578	0.575
Gujarati	0.359	0.417	0.446
Punjabi	0.264	0.337	0.198
Marathi	0.379	0.383	0.390
Hin-RG63			
Hindi	0.685	0.696	0.624

Table 3: Performance (*Spearman correlation*) of models trained on the complete monolingual corpus with default hyperparameters.

Language	**PPMI**	**SVD**	**SGNS**
WS-235			
Hindi	0.566	0.609	0.550
Gujarati	0.343	0.483	0.461
Punjabi	0.259	0.342	0.210
Marathi	0.361	0.421	0.369
Hin-RG63			
Hindi	0.703	0.704	0.548

Table 4: Performance (*Spearman correlation*) of models trained on the complete monolingual corpus with hyperparameters as recommended in (Levy et al., 2015)

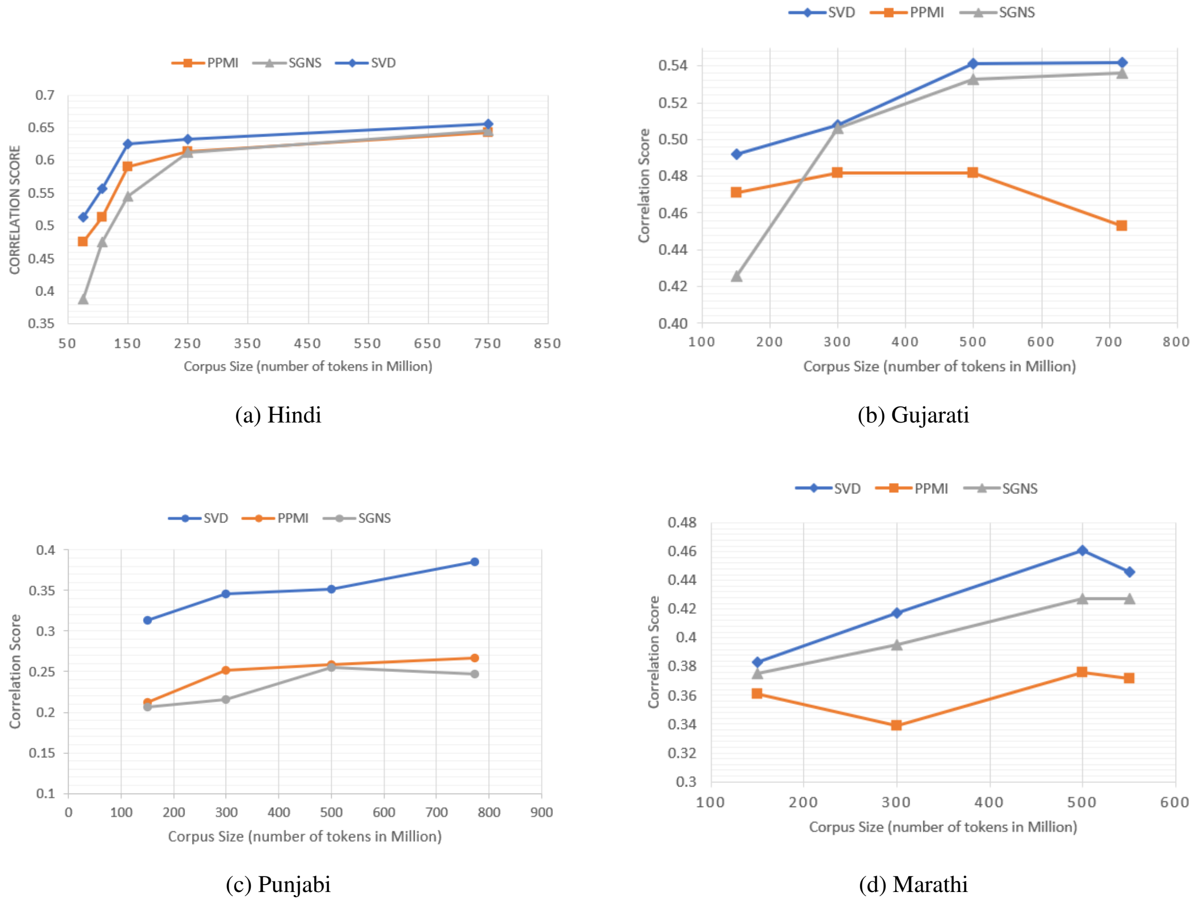
4.3.1 Full Corpus

We train the investigated distributional semantic methods on the complete corpora for each language. Table 3 and Table 4 report the performance of models trained with default hyperparameters and configuration recommended for English respectively. On comparing the performance with models trained on optimal hyperparameters for Hindi, we see a superior performance for each model across all the Indo-Aryan languages (Table 5). This result demonstrates that hyperparameter configurations can be adapted well for closely-related languages.

We further compare our fine-tuned models with two pre-trained word embedding models, namely fastText (FT-WC), trained on Common Crawl and Wikipedia (Grave et al., 2018) and IndicFastText (I-FT) (Kakwani et al., 2020). We note that for Hindi, IndicFastText is trained on a *larger* dataset than

Language	**PPMI**	**SVD**	**SGNS**	**FT-WC**	**I-FT**
WS-235					
Hindi	0.642	**0.656**	0.645	0.550	0.598
Gujarati	0.453	0.542	0.536	0.477	**0.567**
Punjabi	0.267	**0.385**	0.247	0.350	0.357
Marathi	0.372	0.446	0.427	**0.464**	0.459
Hin-RG63					
Hindi	0.722	**0.809**	0.724	0.758	0.632

Table 5: Performance (*Spearman correlation*) of models trained on the complete monolingual corpus with optimal hyperparameters for Hindi. FT-WC is fastText trained on Wikipedia and Common Crawl; I-FT is IndicFastText.



(a) Hindi

(b) Gujarati

(c) Punjabi

(d) Marathi

Figure 1: Performance of PPMI, SVD and SGNS on WS-235 for Hindi,Gujarati, Punjabi and Marathi. Performance for each model is reported for varying sizes of the corpus.

the previously released fastText. Table 5 shows the evaluation results for all five models. Out of the three studied models, SVD consistently shows superior performance over all four languages. On comparison with FastText and IndicFastText, our modified SVD either outperforms both or is on par. For Hindi, since the tuning is exhaustive we observe the three tuned models to achieve a higher score than the FastText models on both Hin-WS235 and Hin-RG63. Here, it should also be noted that the training corpus used to achieve this score was almost half the size of the one used by IndicFastText, further supporting our rationale of fine-tuning. Our SVD model with adapted hyperparameters for the other Indo-Aryan languages performs on par with the fastText models, even outperforms IndicFastText for Punjabi. We also confirmed that the current dimensionality settings did not affect our results for SGNS, as experiments with lower dimensionalities of 200 and 300 showed negligible gains for SGNS.

4.3.2 Low-resource setting

We analyze the performance of all three models with varying data sizes for each language. We would like to experiment with other Indo-Aryan languages which are truly low-resource, but evaluation datasets only exist for a fraction of available languages. Hence, we decided to experiment with the same languages using different sizes.

The best correlation score of each method after training on different slices of corpora is shown in Figure 1. We infer from the graphs that SVD is quite robust with respect to data size and even with a fraction of data i.e. in case of low-resource scenario, does not show a considerable dip in performance. For Punjabi, there is a substantial gain in performance when using SVD across all data sizes. On the other hand, SGNS and SVD are almost similar for Gujarati, and SGNS may improve with more data.

For Hindi, careful tuning of SVD and PPMI even on a fifth of the complete corpus attains a performance on par with FastText and IndicFastText.

5 Summary

Our work shows that careful hyperparameter tuning can go a long way in improving the performance of distributed semantic models. Interestingly, we find that the general recommendations in Levy et al. (2015) for hyperparameter settings work well only for English word representations. For Hindi, some of their recommendations hold, whereas others do not. Moreover, the hyperparameter settings for Hindi carry over well to related languages and there is performance improvement compared to the default or English-specific settings. This seems to suggest that language specific differences are playing a role with respect to hyperparameter settings.

Perhaps the most interesting result is that modified SVD is more robust than SGNS, both across languages and data sizes. This contrasts with the idea that the differences between the architectures of particular distributional semantic models are trivial so long as they are trained in a similar fashion (Levy et al., 2015).

We also note that the problem of building word embeddings for low-resource languages has been addressed using cross-lingual representations (Ruder et al., 2019). These rely on alignments between low-resource and better-resourced languages. Techniques such as cognate detection have been used to improve these alignments (Sharoff, 2020). Newer contextualized word embedding models can make these alignments internally, allowing for cross lingual transfer (Conneau et al., 2020). In this paper we have chosen to focus on monolingual word embeddings and leave the exploration of cross-lingual representations for future work.

Acknowledgements

The authors acknowledge the Summer Fellowship Research programme (SFRP-2020) of CEP, IIT Delhi, which enabled Kushagra Bhatia and Divyanshu Aggarwal to pursue research in IIT Delhi.

References

Syed Sarfaraz Akhtar, Arihant Gupta, Avijit Vajpayee, Arjit Srivastava, and Manish Shrivastava. 2017. Unsupervised morphological expansion of small datasets for improving word embeddings. In *Proceedings of the 18th International Conference on Computational Linguistics and Intelligent Text Processing (CICLING)*.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland. Association for Computational Linguistics.

Ondrej Bojar, Vojtech Diatka, Pavel Rychlỳ, Pavel Stranák, Vít Suchomel, Ales Tamchyna, and Daniel Zeman. 2014. Hindencorp-hindi-english and hindi-only corpus for machine translation. In *LREC*, pages 3550–3555.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. A framework for the construction of monolingual and cross-lingual word similarity datasets. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 2: Short papers)*, pages 1–7.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the nlp world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

David Jurgens, Mohammad Taher Pilehvar, and Roberto Navigli. 2014. Semeval-2014 task 3: Cross-level semantic similarity. In *SemEval@ COLING*, pages 17–26.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Herbert Rubenstein and John B Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *J. Artif. Int. Res.*, 65(1):569–630.

Magnus Sahlgren and Alessandro Lenci. 2016. The effects of data size and frequency range on distributional semantic models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 975–980, Austin, Texas. Association for Computational Linguistics.

Serge Sharoff. 2020. Finding next of kin: Cross-lingual embedding spaces for related languages. *Natural Language Engineering*, 26(2):163–182.