Hubness Reduction Improves Sentence-BERT Semantic Spaces

Beatrix M. G. Nielsen*[1] and Lars Kai Hansen[1]

[1]Technical University of Denmark
{bmgi, lkai}@dtu.dk

Abstract

Semantic representations of text, i.e. representations of natural language which capture meaning by geometry, are essential for areas such as information retrieval and document grouping. High-dimensional trained dense vectors have received much attention in recent years as such representations. We investigate the structure of semantic spaces that arise from embeddings made with Sentence-BERT and find that the representations suffer from a well-known problem in high dimensions called hubness. Hubness results in asymmetric neighborhood relations, such that some texts (the hubs) are neighbours of many other texts while most texts (so-called anti-hubs), are neighbours of few or no other texts. We quantify the semantic quality of the embeddings using hubness scores and error rate of a neighbourhood based classifier. We find that when hubness is high, we can reduce error rate and hubness using hubness reduction methods. We identify a combination of two methods as resulting in the best reduction. For example, on one of the tested pretrained models, this combined method can reduce hubness by about 75% and error rate by about 9%. Thus, we argue that mitigating hubness in the embedding space provides better semantic representations of text.

1 Introduction

Large Language Models (LLMs) like BERT [1] have been shown to be effective for information retrieval (e.g. [2], [3]) and using the siamese network construction, the Sentence-BERT architecture [4] has made it possible to create semantically meaningful text embeddings effectively enough to make large-scale text clustering viable. Sentence-BERT can be viewed as a fine tuning of a LLM to produce semantically relevant sentence embeddings. We explore the geometry of the resulting embedding space using the neighbourhoods of the data points.

For a typical Sentence-BERT setup, embeddings are of dimension 768. In such high dimensional data, so-called hubs tend to occur (e.g., [5], [6]). Hubs are points which are among the k nearest neighbours of many other points even for small k. Antihubs, which are in the nearest neighbours of few or no other points, also tend to appear in large numbers when the dimension is high ($\gtrsim 10$). These hubs and antihubs skew the nearest neighbour relation, since if a point, x, is in the ten nearest neighbours of a hundred other points, most of those points cannot be in the ten nearest neighbours of x. Therefore, if we use the neighbour relation to define semantic similarity, we get the counter intuitive result, that some texts are semantically similar to many other texts while many texts are not semantically similar to anything. Hubness has been found in many different kinds of data and degrades performance of neighbourhood based algorithms whether they are used for classification [6], regression [7], clustering (e.g. [8], [9]) or outlier detection [10]. Hubness has also been found to cause vulnerability in recommender systems [11]. Naturally, there has been a quest in the cited literature for means to mitigate the effect of hubness.

In this paper, we consider differently trained Sentence-BERT models and explore the effect on hubness scores and error rates of K-Nearest Neighbours (knn) classification when using post hoc hubness reduction methods. The overall picture is that the methods do reduce hubness and in most cases also reduce error rate, i.e., increase neighbor semantic similarity. For example, for a medium sized base model trained using cosine similarity and no normalisation we get a reduction of mean error rate over 12 seeds between 7% and 13% on the three used datasets. We also consider four pretrained models, trained on more data than our own and show that for the 20 Newsgroups data set, we can reduce hubness between 69% and 83% and reduce error rates between 7% and 9%, i.e. training on more data does not solve the hubness problem.

Our main contributions are:

- Empirical evidence that high hubness can occur in Sentence-BERT embeddings, even when the model has been trained on large amounts of data.
- A new use for a parameter free transformation of data (f-norm) as a hubness reduction method. This method forces the dimensions of embeddings to follow a standard normal distribution and has been used before by [13], but never to our knowledge for hubness reduction.
- Experimental results which show the benefits of

[*] Corresponding Author.

Proceedings of the 5th Northern Lights Deep Learning Conference (NLDL), PMLR 233, 2024.

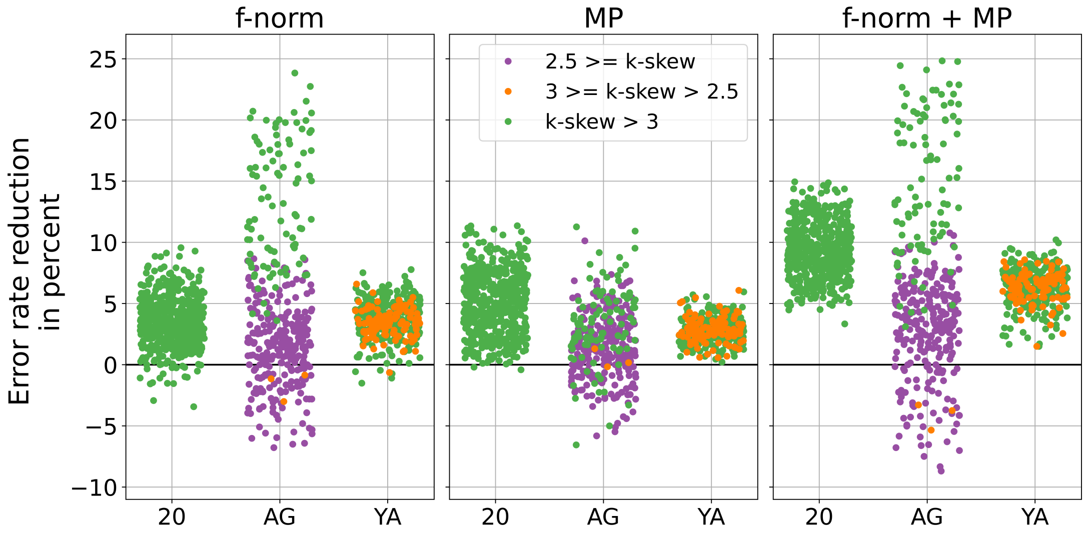


Figure 1. Error rate reduction for our models when using non-parametric standardization of the histogram (f-norm), mutual proximity (MP) as introduced by [12], and the combination of f-norm + MP. Note that the base error rate varies for the models. This figure only shows the changes. If we only consider models where base embeddings have a k-skewness of more than 3, f-norm + MP reduces error rate in all cases. If we only consider embeddings where k-skewness is more than 2.5, f-norm + MP reduces error rate in all except 3 cases. Datasets: 20 Newsgroups (20), AG News (AG), 10% of Yahoo Answers (YA).

three hubness reduction methods f-norm, Mutual Proximity (MP) [12] and f-norm + MP and identify f-norm + MP as the best method for hubness and error rate reduction on three datasets. See figure 1 for error rate reduction and figures 3, 4 for hubness reduction.

1.1 Related Work

In Radovanovic et al. [14] and [6], they described why hubs appear in high dimensional data and illustrated the hubness problem on both synthetic and real data. [15] reviewed hubness reduction methods for a large variety of datasets. They concluded that e.g. local scaling [16] and mutual proximity [12] formed state of the art in hubness reduction. Both [6] and [15] included text datasets in their analyses. However, both works are based on very high dimensional and sparse bag-of-word-representations. Hubness has also been found in word embeddings e.g. [17]. [18] used hubness reduction to improve performance in zero-shot translation and image labelling.

Recently, [19] introduced query bank normalisation to mitigate the hubness problem when using cross modal embeddings to retrieve images. Z-score normalisation has been used to reduce hubness of image embeddings [20] and improve performance for few-shot learning. [21] used the average of the context embeddings in the last layer of BERT as sentence embeddings, and showed that when using cosine similarity, accuracy of semantic similarity on text could be improved by transforming the distribution of the embeddings to a smooth isotropic normal distribution using normalizing flows. This idea is similar to our idea of making dimensions standard normally distributed, however, our method does not require training. [22] consider the problem of ”rogue dimensions”, a few dimensions of an embedding which are far from the origin, have high variance and dominate the cosine similarity, so that these few dimensions decide whether two embeddings are similar or not, even though they do not have much say in model decisions. See section 5 for further relations to our work.

2 Hubness

Let $X = (x_1, ..., x_m) \subset \mathbb{R}^D$ be m points in D-dimensional space. We define $N_k(x)$, called the k-occurrence, for a point $x \in X$ as the number of times x occurs in the k nearest neighbours of all points in X, for some distance measure. In this paper we will use Euclidean distance. We will use the k-occurrence, $N_k(x)$, to measure the hubness of our embedding spaces. However, for comparing several types of embeddings it is more convenient with a single score, so we will use the skewness of the $N_k(x)$ distribution (k-skewness) as proposed in [6] and the robinhood score (rh) as proposed in [23]. The robinhood score, H^k, is given by

$$H^k = \frac{\sum_{x \in X} |N_k(x) - k|}{2k(m-1)} \tag{1}$$

and measures how far the k-occurrences of the embeddings are from the expected k. That is, it is a fraction between 0 and 1, which tells us how large

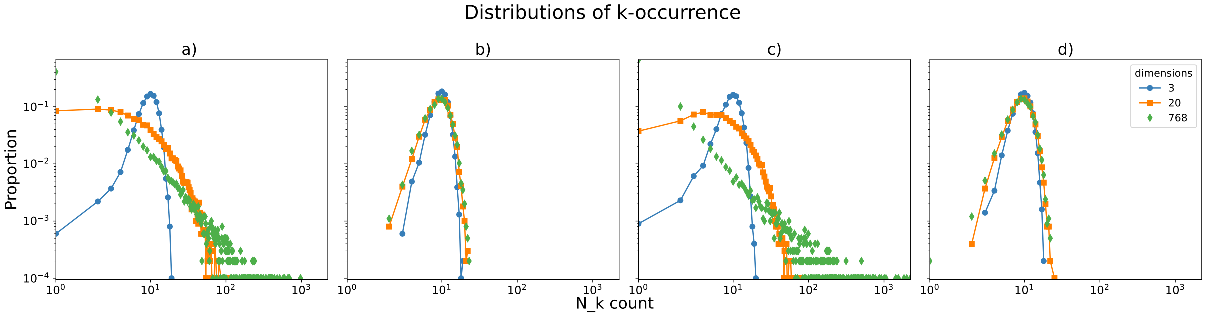


Figure 2. Using 10 neighbours. Data with dimensions a) standard normally distributed (k-skewness: −0.10, 2.32 and 11.62, rh: 0.09, 0.35 and 0.61), b) standard normally distributed with embeddings normalized to unit length (k-skewness: 0.05, 0.27 and 0.37, rh: 0.08, 0.11 and 0.12), c) F distributed (k-skewness: −0.12, 1.78 and 19.30, rh: 0.10, 0.28 and 0.74) and d) F distributed after using f-norm (k-skewness: 0.04, 0.34 and 0.34, rh: 0.9, 0.11 and 0.12). b and d seem stable to high dimension.

a part of nearest neighbours needs to be changed if all objects are to be in the k nearest neighbours of the same number of points. We use the scikit-hubness library by [24] to calculate these scores. In this paper, we use $k = 10$ for measuring k-skewness and robinhood score, since the exact value is not important and 10 is the value used in much of the literature we reference.

Algorithm 1 f-norm

```
$X = \{x_{i,d}\}_{i \in \{1,\dots,m\}, d \in \{1,\dots,D\}}$
for $d \in \{1, \dots, D\}$ do
    $Z \leftarrow (z_1, \dots, z_m) \sim \mathcal{N}(0,1)$
    $idxData \leftarrow \text{argsort}_{i \in \{1,\dots,m\}}(x_{i,d})$
    $idxNormal \leftarrow \text{argsort}_{i \in \{1,\dots,m\}}(z_i)$
    for $idx_1, idx_2 \in (idxData, idxNormal)$ do
        $x_{idx_1,d} \leftarrow z_{idx_2}$
    end for
end for
for $i \in \{1, \dots, m\}$ do
    $x_i \leftarrow \frac{x_i}{\|x_i\|_2}$
end for
```

According to [6], in unimodal data distributions there is a strong correlation between $N_k(x)$ and how close x is to the global data mean. That is, points close to the center of the data tend to become hubs. Therefore, we expect hubness to be lower when all points have equal distance to the mean, which will be the case if for example all points are on the surface of a hypersphere (more in appendix B). Which means that if our data is standard normally distributed in each dimension, and we then normalise to unit length, we should get lower hubness. In figure 2a) and b) we see that this is indeed the case. In LLM derived embeddings we do not expect the individual dimensions to be distributed according to the same normal distribution, however, we find empirically that the distributions of the dimensions of sentence BERT embeddings do look close to normal distributions (examples in Appendix figure F.1). See more motivation in Appendix D. Thus, we explored the idea to force the dimensions of the data points to follow a standard normal distribution, but preserving the ranking of the values (we call this method 'f-norm', Algorithm 1). Note, when using this method the distribution of the dimensions before the transformation no longer matters. See figure 2c) and d) for an example with F distributed dimensions. See figure 3 and appendix figure E.1 for how f-norm affects some real data.

2.1 Methods for hubness reduction of semantic spaces

Normalisations used during training aim to center the data and/or make it lie on the unit sphere, see Appendix D. In our models we used: No normalisation ('none'), normalisation of embeddings to unit length ('n'), centering of embeddings (subtracting the mean value of the embedding from all dimensions in the embedding) ('c'), a combination of centering and normalisation of embeddings to unit length ('c,n') and z-score normalisation of embeddings (according to [20] this had an effect when doing few shot learning) ('z').

After training and obtaining embeddings for a dataset using a model, we tested three different post hoc methods for hubness reduction on the three datasets.

- Forcing standard normal distribution in each dimension of embeddings and then normalising embeddings to unit length (f-norm) as described in Algorithm 1.
- Mutual proximity (MP) as introduced by [12].
- First forcing a standard normal distribution in each dimension and normalising embeddings, then using mutual proximity on the result (f-norm + MP).

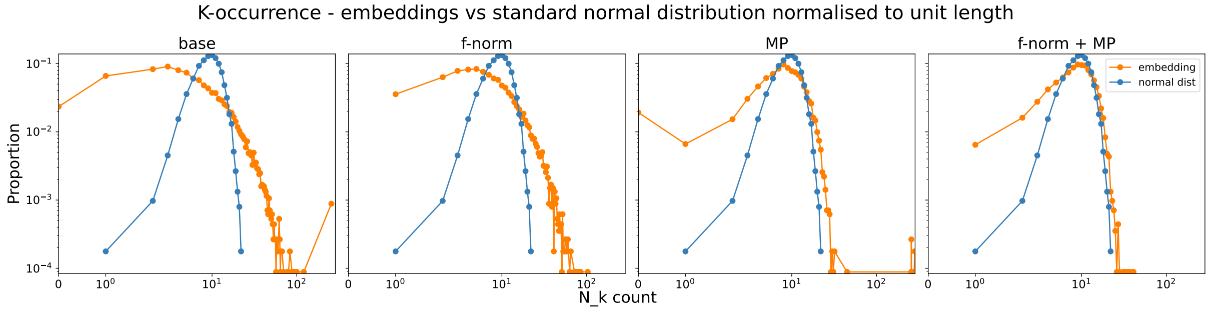


Figure 3. K-occurrence distribution (10 neighbours) for embeddings on 20 newsgroups (orange) compared to points coming from a standard normal distribution in same number of dimensions (blue) with embeddings normalised to unit length. Using our reduction methods, the k-occurrence of the embeddings come closer to the stable distribution. Note MP still leaves some large hubs. Embedding k-skewness: 8.79, 2.20, 20.98, 0.42, rh: 0.35, 0.30, 0.20, 0.16. Model: distilroberta-base_euclidean_n_seed1.

On the 20 Newsgroups dataset, we tested two additional methods: f-uniform and local scaling. The latter was introduced by [16] and proposed for hubness reduction by [12]. More in Appendix C.

MP aims to make the nearest neighbour relation more symmetric by translating the distance between points $d(x, y)$ to a probability $MP(d_{x,y})$ that x is the nearest neighbour of y and that y is also a nearest neighbour of x in the following way: Let $p(d_x)$ be the probability density function of the probability distribution of distances from x to all other points in the dataset. Then use $p(d_x)$ and $d(x, y)$ to find the probability of y being the nearest neighbour of x. Then use $p(d_y)$ and $d(x, y)$ to find the probability of x being the nearest neighbour of y. It is then these probabilities which are used as the distance in the knn algorithm. [12] showed that it is enough to calculate distances to a sample of other points. More in Appendix P.

3 Evaluation

We trained Sentence-BERT models using three different base models, which were not trained on the datasets we use for testing, (microsoft-MiniLM-L12-H384-uncased[1] [25], distilroberta-base[2] [26], microsoft-mpnet-base[3] [27]) and three different distance measures during training (cosine similarity ('cos'), cosine distance ('cos_dist'), Euclidean distance ('euclidean')). For each combination of model, distance measure and normalisation method from 2.1, we trained with twelve random seeds (model naming scheme is [base model]_[distance measure]_[normalisation]_[seed]). This gave us 540 models in all. On the large datasets (AG News and Yahoo Answers), we only calculated embeddings and results for models using the medium and large base models, that is distilroberta-base and microsoft-mpnet-base, so on these datasets we have results for 360 models. We trained our models on the STS benchmark dataset[4] for 4 epochs. More in Appendix J, K, L.

We also made embeddings using four pretrained Sentence-BERT models, which have been trained on more data than our own. These models were: all-mpnet-base-v2[5], multi-qa-distilbert-cos-v1[6], all-distilroberta-v1[7] and all-MiniLM-L12-v2[8]. Note that all these pretrained models have been trained on Yahoo Answers in various ways. More in Appendix I.

We used three text classification datasets: 20 Newsgroups, AG News (by Antonio Gulli) and ten percent of the Yahoo Answers dataset [28]. Details in Appendix A.

We classify using knn. To identify the number of neighbours to use in knn for classification, we used stratified 10-fold validation on the training set. We measured error rate on the test set. More details in Appendix M.

Let the train embeddings be an $m \times d$ matrix and the test embeddings be an $l \times d$ matrix. When using f-norm at test time, we append the test embeddings to the train embeddings matrix giving us an $(m + l) \times d$ matrix, which we then use the algorithm on.

4 Results

Trained, models, embeddings and raw result files can be found at DTU Data[9]. Code can be found

[1] huggingface.co/MiniLM-L12-H384-uncased
[2] huggingface.co/distilroberta-base
[3] huggingface.co/microsoft/mpnet-base
[4] sbert.net/datasets/stsbenchmark.tsv.gz
[5] huggingface.co/sentence-transformers/all-mpnet-base-v2
[6] huggingface.co/sentence-transformers/multi-qa-distilbert-cos-v1
[7] huggingface.co/sentence-transformers/all-distilroberta-v1
[8] huggingface.co/sentence-transformers/all-MiniLM-L12-v2
[9] https://doi.org/10.11583/DTU.c.6165561.v1

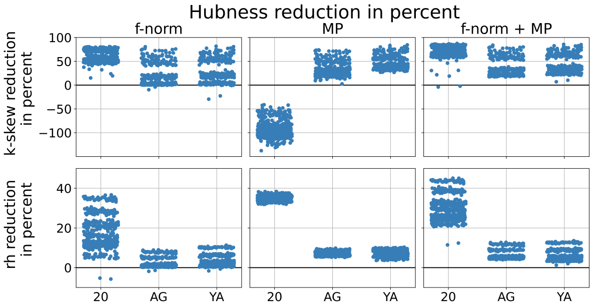


Figure 4. Reduction in hubness using f-norm, MP and f-norm + MP measured with k-skewness (k-skew) and robinhood score (rh). Higher is better. Hubness is reduced in almost all cases, however for MP on 20 Newsgroups hubness increases as measured with k-skewness. Likely induced by the hubs remaining after MP, see figure 3.

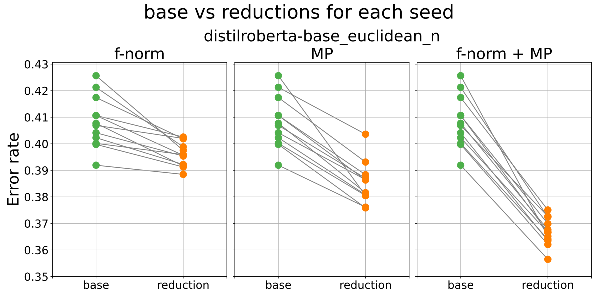


Figure 5. knn error on 20 Newsgroups before vs after hubness reductions. Average base k-skewness 6.06. See 3

on GitHub[10]. Overviews of training combinations and post hoc reduction methods with error rate and hubness scores are in Appendix S. See this material for results on f-uniform and local scaling. Summary of error rate changes from the overview tables is in figure 1, summary of the changes in hubness is in figure 4.

We see that the methods reduce hubness as measured with the robinhood score in almost all cases (figure 4). However, when using MP, k-skewness went up in all cases on 20 Newsgroups. Inspecting the k-occurrence distribution, this seems to be because while MP makes the knn relation more symmetric for most data points, there are still some rather large hubs which it does not change. These hubs are sentences which are either empty or which only contain whitespace characters. In 20 Newsgroups there are 268 sentences which are either empty or whitespace. Before hubness reduction, 252 of them have a k-occurrence of 0 and 10 have a k-occurrence of more than 100. After using MP, 191 of them have a k-occurrence of 0 and 10 still have a k-occurrence of more than 100. However, when using f-norm + MP there are no longer these very large hubs (figure 3), in fact after f-norm + MP, all the empty or whitespace sentences have k-occurrences between 4 and 17. Thus, these sentences are no longer disconnected with respect to the neighbourhood relation. Using f-norm + MP also decreases k-skewness in more cases than using just f-norm.

On all three datasets, when k-skewness is more than 3, error rate is reduced when using f-norm + MP (figure 1). See figure 5 for examples of absolute error reduction. More examples in Appendix G. Embeddings on AG News had lower base hubness and error rate than embeddings on the other two datasets, however, since f-norm reduces error rate for all embeddings with k-skewness more than 3, we think it likely that the low hubness makes it difficult for the hubness reduction methods to improve classification performance on this dataset. We also found that f-norm + MP is more likely to have a larger effect on error rate, if the base skewness is higher (Appendix Q). We hypothesize that when hubness is not high enough, the transformation of the embeddings harms the learned structure of the embeddings more than it helps.

The lowest mean error rates came from models using the largest base model. However, using f-norm + MP can in some cases make up for the model size. For example, distilroberta-base_cos_none base and microsoft-mpnet-base_cos_none had base mean error of 10.9% and 9.4% on AG News. Using f-norm + MP reduces the error to 8.8% and 8.4%. So the smaller model with f-norm + MP is better than the bigger model without. More about the smallest error rates in Appendix N.

We found that models which use some kind of embedding length normalisation (n; c,n; z) tend to have slightly lower hubness measured with robinhood score than models which do not have this kind of normalisation (none; c). More on differences between normalisations during training in Appendix O. We found that which distance measure is used during training is not a good predictor for error rate or effectiveness of hubness reduction. On AG News, the ten best mean error rates all came from models which used Euclidean distance for training, however on 20 Newsgroups, models trained with Euclidean distance had worse error rate than models trained with cosine similarity or cosine distance.

4.1 Pretrained Models

All the tested pretrained models have a high base hubness as measured with k-skewness on 20 Newsgroups (See Table 1). This means that training models on more data will not necessarily result in low hubness. On the 20 Newsgroup dataset, the error rate goes down (reduction between 7% and 9%.) when using f-norm + MP for all four models. These reductions are highly significant, since

[10] https://github.com/bemigini/hubness-reduction-improves-sbert-semantic-spaces

Table 1. Error rate (error) and k-skewness (skew) on 20 Newsgroups (20), AG News (AG) and ten percent of Yahoo Answers (YA) for four pretrained models, base case vs after hubness reduction. Note that all these models have been trained on YA in various ways and their hubness on YA is in general lower than the models we trained. Thus, it seems that training on a specific dataset can make hubness go down for that dataset. However, hubness is still high on 20 which the models have not been trained on.

Model		**20**		**AG**		**YA**	
		error	**skew**	**error**	**skew**	**error**	**skew**
multi-qa-distilbert-cos-v1	**base**	0.318	7.58	0.081	1.90	0.271	2.12
	f-norm + MP	0.289	1.89	0.079	1.57	0.269	1.88
all-MiniLM-L12-v2	**base**	0.324	11.45	0.088	1.91	0.268	1.98
	f-norm + MP	0.300	1.88	0.085	1.56	0.267	1.79
all-mpnet-base-v2	**base**	0.295	11.53	0.082	1.81	0.251	2.02
	f-norm + MP	0.272	1.98	0.082	1.54	0.250	1.78
all-distilroberta-v1	**base**	0.322	6.61	0.085	1.93	0.254	2.08
	f-norm + MP	0.296	2.05	0.085	1.59	0.253	1.82

using McNemar's test gives p-values smaller than $2.1 \cdot 10^{-9}$. However, error rate is not reduced for the other two datasets. We interpret this as being due to the low base hubness of the pretrained models on AG News and Yahoo Answers. AG News also had a low base hubness for most of our own models, so we suppose this is due to the nature of the dataset. Our own models had higher base hubness on Yahoo Answers than the pretrained models, however all the pretrained models were also trained on Yahoo Answers in various ways. Therefore, we assume that training on a specific dataset can make hubness go down for that dataset. See Table I.1 in Appendix I for more details.

5 Discussion and Conclusion

Since f-norm also reduces the rogue dimensions problem [22], the reductions in error might not be solely due to reduced hubness. However, in cases where MP reduces error rate, we hypothesize that hubness reduction does contribute to the error reduction.

We measured hubness with both k-skewness and the robinhood score. The robinhood score is less sensitive to outliers, so it can detect when a method reduces the overall hubness even though it leaves some outliers, while k-skewness might grow in such cases. However, k-skewness is a better predictor for whether hubness reduction methods using f-norm will have a positive effect on the error rate. The robinhood score has a different baseline scale for each dataset, hampering its interpretation.

We have only looked at how hubness reduction methods affect knn. Since we used Euclidean distance in our knn, the error rates on our datasets tell us how well the Euclidean distances between datapoints correspond to class structure. Thus, when searching for other methods which might benefit from these hubness reduction methods, we should look for models which use distances between the datapoints for classification. Other clustering or classification techniques might not be improved using these methods (Note on SVM in Appendix R). When using f-norm for m train embeddings and l test embeddings of dimension d, one will have to sort $(m+l) \cdot 2d$ numbers. However, for very large training datasets, it should not be necessary to use all the training data. It would be an interesting direction for further research to find out how many and which training points to keep to get good performance.

In conclusion, when using neighbourhood based approaches and high dimensional text embeddings, one should always check for hubness. For a broad set of training choices, if there is high hubness in embeddings (k-skewness of more than 3 seems to be useful as a rule of thumb), then hubness reduction techniques (especially f-norm + MP) will produce better text representations as shown by more symmetric nearest neighbour relation and improved classification.

Acknowledgements

This work was supported by the Danish Pioneer Centre for AI, DNRF grant number P1 and by the Novo Nordisk Foundation grant NNF22OC0076907 "Cognitive spaces - Next generation explainability". Thanks to Mikkel N. Schmidt for helpful comments on the abstract and introduction.

References

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Confer-*

ence of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://aclanthology.org/N19-1423.

[2] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. "Dense Passage Retrieval for Open-Domain Question Answering". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 6769–6781. DOI: 10.18653/v1/2020.emnlp-main.550. URL: https://aclanthology.org/2020.emnlp-main.550.

[3] Z. Jiang, A. El-Jaroudi, W. Hartmann, D. Karakos, and L. Zhao. "Cross-lingual Information Retrieval with BERT". English. In: *Proceedings of the workshop on Cross-Language Search and Summarization of Text and Speech (CLSSTS2020)*. Marseille, France: European Language Resources Association, May 2020, pp. 26–31. ISBN: 979-10-95546-55-9. URL: https://aclanthology.org/2020.clssts-1.5.

[4] N. Reimers and I. Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. DOI: 10.18653/v1/D19-1410. URL: https://aclanthology.org/D19-1410.

[5] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. "When Is "Nearest Neighbor" Meaningful?" In: *Database Theory — ICDT'99*. Ed. by C. Beeri and P. Buneman. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 217–235. ISBN: 978-3-540-49257-3. URL: https://www.researchgate.net/publication/2845566_When_Is_Nearest_Neighbor_Meaningful.

[6] M. Radovanovic, A. Nanopoulos, and M. Ivanovic. "Hubs in space: Popular nearest neighbors in high-dimensional data". In: *Journal of Machine Learning Research* 11.sept (2010), pp. 2487–2531. URL: https://dl.acm.org/doi/10.5555/1756006.1953015.

[7] K. Buza, A. Nanopoulos, and G. Nagy. "Nearest neighbor regression in the presence of bad hubs". In: *Knowledge-Based Systems* 86 (2015), pp. 250–260. DOI: https://doi.org/10.1016/j.knosys.2015.06.010.

[8] D. Schnitzer and A. Flexer. "The unbalancing effect of hubs on k-medoids clustering in high-dimensional spaces". In: *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2015, pp. 1–8. DOI: 10.1109/IJCNN.2015.7280303.

[9] N. Tomasev, M. Radovanovic, D. Mladenic, and M. Ivanovic. "The role of hubness in clustering high-dimensional data". In: *IEEE transactions on knowledge and data engineering* 26.3 (2013), pp. 739–751. DOI: 10.1109/TKDE.2013.25.

[10] M. Radovanović, A. Nanopoulos, and M. Ivanović. "Reverse nearest neighbors in unsupervised distance-based outlier detection". In: *IEEE transactions on knowledge and data engineering* 27.5 (2014), pp. 1369–1382. DOI: 10.1109/TKDE.2014.2365790.

[11] K. Hara, I. Suzuki, K. Kobayashi, and K. Fukumizu. "Reducing hubness: A cause of vulnerability in recommender systems". In: *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 2015, pp. 815–818.

[12] D. Schnitzer, A. Flexer, M. Schedl, and G. Widmer. "Local and global scaling reduce hubs in space". In: *Journal of Machine Learning Research* 13.10 (2012). URL: https://jmlr.csail.mit.edu/papers/volume13/schnitzer12a/schnitzer12a.pdf.

[13] T. J. Abrahamsen and L. K. Hansen. "A Cure for Variance Inflation in High Dimensional Kernel Principal Component Analysis". In: *Journal of Machine Learning Research* 12.6 (2011). URL: https://www.jmlr.org/papers/volume12/abrahamsen11a/abrahamsen11a.pdf.

[14] M. Radovanović, A. Nanopoulos, and M. Ivanović. "Nearest neighbors in high-dimensional data: The emergence and influence of hubs". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. 2009, pp. 865–872. DOI: https://doi.org/10.1145/1553374.1553485.

[15] R. Feldbauer and A. Flexer. "A comprehensive empirical comparison of hubness reduction in high-dimensional spaces". In: *Knowledge and Information Systems* 59.1 (2019), pp. 137–166. DOI: https://doi.org/10.1007/s10115-018-1205-y.

[16] L. Zelnik-Manor and P. Perona. “Self-tuning spectral clustering”. In: *Advances in neural information processing systems* 17 (2004). URL: https://proceedings.neurips.cc/paper/2004/file/40173ea48d9567f1f393b20c855bb40b-Paper.pdf.

[17] A. Ormazabal, M. Artetxe, G. Labaka, A. Soroa, and E. Agirre. “Analyzing the Limitations of Cross-lingual Word Embedding Mappings”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 4990–4995. DOI: 10.18653/v1/P19-1492. URL: https://aclanthology.org/P19-1492.

[18] A. Lazaridou, G. Dinu, and M. Baroni. “Hubness and Pollution: Delving into Cross-Space Mapping for Zero-Shot Learning”. 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)*. Beijing, China: Association for Computational Linguistics, July 2015, pp. 270–280. DOI: 10.3115/v1/P15-1027. URL: https://aclanthology.org/P15-1027.

[19] S.-V. Bogolin, I. Croitoru, H. Jin, Y. Liu, and S. Albanie. “Cross Modal Retrieval with Querybank Normalisation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 5194–5205. DOI: https://doi.org/10.48550/arXiv.2112.12777.

[20] N. Fei, Y. Gao, Z. Lu, and T. Xiang. “Z-score normalization, hubness, and few-shot learning”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 142–151. URL: https://openaccess.thecvf.com/content/ICCV2021/papers/Fei_Z-Score_Normalization_Hubness_and_Few-Shot_Learning_ICCV_2021_paper.pdf.

[21] B. Li, H. Zhou, J. He, M. Wang, Y. Yang, and L. Li. “On the Sentence Embeddings from Pre-trained Language Models”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 9119–9130. DOI: 10.18653/v1/2020.emnlp-main.733. URL: https://aclanthology.org/2020.emnlp-main.733.

[22] W. Timkey and M. van Schijndel. “All Bark and No Bite: Rogue Dimensions in Transformer Language Models Obscure Representational Quality”. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4527–4546. DOI: 10.18653/v1/2021.emnlp-main.372. URL: https://aclanthology.org/2021.emnlp-main.372.

[23] R. Feldbauer, M. Leodolter, C. Plant, and A. Flexer. “Fast approximate hubness reduction for large high-dimensional data”. In: *2018 IEEE International Conference on Big Knowledge (ICBK)*. IEEE. 2018, pp. 358–367. DOI: 10.1109/ICBK.2018.00055.

[24] R. Feldbauer, T. Rattei, and A. Flexer. “scikit-hubness: Hubness Reduction and Approximate Neighbor Search”. In: *Journal of Open Source Software* 5.45 (2020), p. 1957. ISSN: 2475-9066. DOI: 10.21105/joss.01957.

[25] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou. “Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 5776–5788. DOI: https://doi.org/10.48550/arXiv.2002.10957.

[26] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”. In: *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing @ NeurIPS 2019*. 2019. DOI: https://doi.org/10.48550/arXiv.1910.01108. arXiv: 1910.01108. URL: https://www.emc2-ai.org/assets/docs/neurips-19/emc2-neurips19-paper-33.pdf.

[27] K. Song, X. Tan, T. Qin, J. Lu, and T.-Y. Liu. “Mpnet: Masked and permuted pre-training for language understanding”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 16857–16867. URL: https://proceedings.neurips.cc/paper/2020/file/c3a690be93aa602ee2dc0ccab5b7b67e-Paper.pdf.

[28] X. Zhang, J. Zhao, and Y. LeCun. “Character-level convolutional networks for text classification”. In: *Advances in neural information processing systems* 28 (2015). DOI: https://doi.org/10.48550/arXiv.1509.01626.

[29] T. Low, C. Borgelt, S. Stober, and A. Nürnberger. “The hubness phenomenon: Fact or artifact?” In: *Towards Advanced Data Anal-*

ysis by Combining Soft Computing and Statistics. Springer, 2013, pp. 267–278. DOI: 10.1007/978-3-642-30278-7_21.

[30] L. Boytsov and B. Naidan. "Engineering efficient and effective non-metric space library". In: *International Conference on Similarity Search and Applications*. Springer. 2013, pp. 280–293. URL: http://boytsov.info/pubs/sisap2013.pdf.

[31] Y. A. Malkov and D. A. Yashunin. "Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs". In: *CoRR* abs/1603.09320 (2016). DOI: https://doi.org/10.48550/arXiv.1603.09320. arXiv: 1603.09320. URL: http://arxiv.org/abs/1603.09320.

Appendix

Table of Contents

A Datasets

We used text classification datasets, where we expect the classes to be related to the semantic meaning of the texts.

- 20 Newsgroups as can be fetched through scikit-learn which fetches data from the 20 newsgroups website[11]. Originally collected by Ken Lang. It consists of 18846 newsgroup posts with 20 different topics split into a train set of 11314 samples and a test set of 7532 samples. Embeddings were made on the posts as given by the scikit-learn library with no additional preprocessing. License: Unknown.
- AG News fetched from Kaggle[12]. Originally made by Antonio Gulli [13]. News stories divided into four classes each with 30,000 training samples and 1,900 testing samples. So in all we have 120,000 training samples and 7,600 test samples. Embeddings were made on the title concatenated with the description with line breaks replaced by spaces. License: Custom. Any non-commercial use allowed.
- Yahoo Answers fetched from Kaggle[14] made by [28]. Questions and best answers from Yahoo Answers from 10 categories. We use 10 percent of this dataset, so for each category we have 14,000 training samples and 600 test samples, which means we have 140,000 training samples and 6000 test samples in all. Indexes of used samples can be found in the code repository[15]. Embeddings were made on the question title concatenated with the question content and the best answer with line breaks replaced by spaces. License: Unknown.

B K-occurrence and the Mean of the Data

According to [6], in unimodal data distributions there is a strong correlation between $N_k(x)$ and how close x is to the global data mean. That is, points close to the center of the data tend to become hubs. While for multimodal distributions [6] found that points close to local cluster means tend to become hubs. [6] also argue that hubness depends on the intrinsic dimension of data and not the dimension of the embedding. [29] points out that high intrinsic

[11] http://qwone.com/~jason/20Newsgroups/

[12] www.kaggle.com/datasets/amananandrai/ag-news-classification-dataset

[13] groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

[14] www.kaggle.com/datasets/yacharki/yahoo-answers-10-categories-for-nlp-csv

[15] https://github.com/bemigini/hubness-reduction-sentence-bert

dimension does not in itself produce high hubness. They argue that high density gradients, e.g., introduced by support boundaries can lead to formation of hubs.

C Additional Hubness Reduction Methods

On the 20 Newsgroups dataset we tested two additional hubness reduction methods:

- Forcing uniform distribution centered at 0 in each dimension of embeddings and then normalising embeddings to unit length (f-uniform) same way as described in section 2.1 "Illustration on Synthetic Data".
- Local scaling as introduced by [16] (Proposed for hubness reduction by [12]).

We tested f-uniform, since it might not be the distribution which is important, but only that the center of the data is the zero vector and that the variance of the dimensions is the same. However, we did expect that f-uniform would destroy the learned structure of the embeddings, since it seems the distribution of the dimensions of the embeddings before hubness reduction is close to normal. We were suprised to find that f-uniform decreased error rate in almost as many cases as f-norm, but we still decided to only continue with f-norm on the two other datasets, because of the observed distribution of the embeddings.

Local scaling introduces a secondary distance measure $LS(d_{x,y})$ between points x, y, which uses a local scaling parameter σ_x - the distance between x and the m'th nearest neighbour of x for a chosen $m \in \mathbb{N}$.

$$LS(d_{x,y}) = 1 - \exp\left(-\frac{d^2_{x,y}}{\sigma_x \sigma_y}\right) \tag{2}$$

We used $m = 5$.

We decided not to continue with local scaling on the two other datasets, since it did not decrease error rate in as many cases as the other hubness reduction methods.

D Illustration on Synthetic Data

Figure 2a) shows the distribution of the k-occurrence for $10,000$ data points drawn from a standard normal distribution in 3, 20 and 768 dimensions (768 is the dimensionality of Sentence-BERT embeddings). Recall that the density function for a D dimensional standard normal distribution is

$$(2\pi)^{-D/2} e^{-\frac{1}{2}\sum_{i=1}^{D} x_i^2} = (2\pi)^{-D/2} e^{-\frac{1}{2}\|x\|^2}. \tag{3}$$

So the distribution is a function of the length of the vector. Hence, by normalising data points to unit length, we obtain a uniform distribution on the hypersphere. Because of [6] we expect hubness to be lower when all points have equal distance to the mean, see appendix B. Note, the distribution of the k-occurrence is symmetric with this normalization even in 768 dimensions, see figure 2b), thus for normal distributed data normalization reduces hubness.

Drawing data points from a normal distribution with a nonzero mean vector, makes the k-occurrence skewed as before even if we normalise embeddings to unit length. However, if we center the data before normalising, we get again a symmetric k-occurrence distribution. If we instead use a uniform distribution (e.g., the interval $[-1, 1]$) for each dimension and again normalise embeddings to unit length, then this k-occurrence will also become symmetric.

However, in LLM derived embeddings we do not expect the individual dimensions to be distributed according to the same normal distribution although empirically, the distributions of the dimensions do look like normal distributions (examples in Appendix figure F.1).

E Spyplot of before and after f-norm

Figure E.1 shows how the number of other sentences a sentence is in the 10 nearest neighbours of is reduced by f-norm.

F Example of Distribution of Dimensions

See figure F.1.

G Error rate - Base vs reductions

Figures G.1, G.2, G.3, G.4, G.5, G.6, G.7, G.8, G.9, G.10, G.11, G.12, G.13, G.14 show examples of how the error rate changes from the base case to cases where hubness reduction methods have been used.

H Error rate - Without vs with Normalisation

Figures H.1, H.2, H.3, H.4, H.5, H.6, H.7 show error rate differences for models which have been trained without and with normalisation to unit length before and after using f-norm + MP.

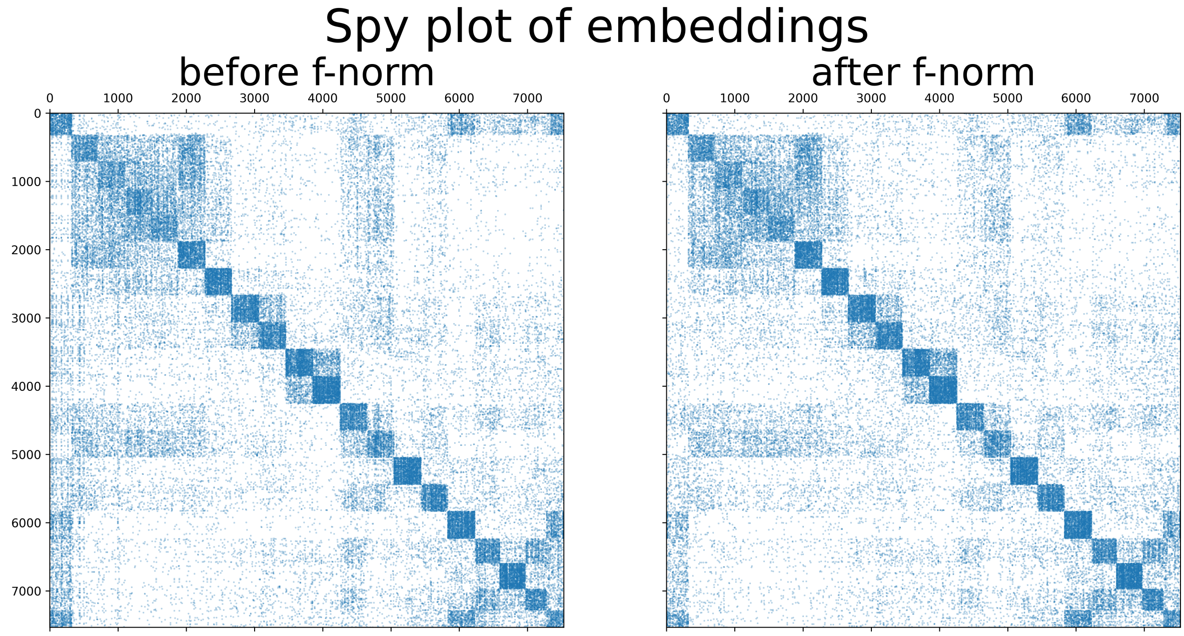


Figure E.1. The model microsoft-MiniLM-L12-H384-uncased_cos_n with seed 1 on 20 Newsgroups test split. Column samples in the 10 nearest neighbours of the row sample are marked with a blue dot, otherwise they are unmarked. Samples are sorted by category. Hubs can be seen as vertical lines especially in the left side of the before plot. These become much less pronounced after having used f-norm.

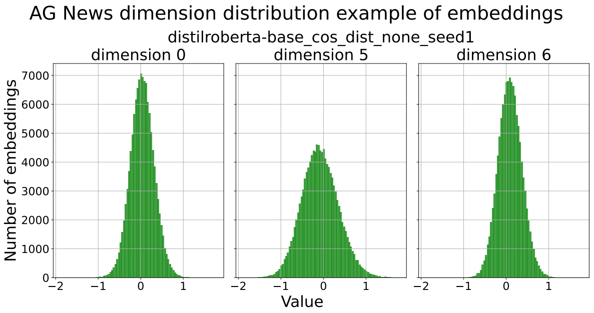


Figure F.1. Values in the dimensions are roughly normally distributed, but do not have the same mean or variance.

I Pretrained Models Results and Intended Use

All pretrained models were made during ”Community week using JAX/Flax for NLP & CV”[16], organized by Hugging Face. All pretrained models used have Euclidean distance as a suitable score function, and they are all supposed to give semantic embeddings of sentences or short paragraphs. Table I.1 shows effects on error rate and hubness when using f-norm, MP and f-norm + MP.

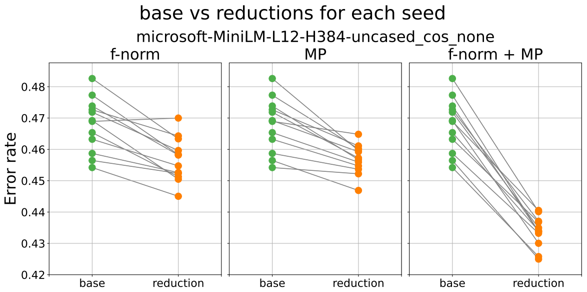


Figure G.1. 20 Newsgroups. Average base k-skewness 8.15.

J Computational Experiment Details

J.1 Hyperparameters

When training our models, we kept the default hyperparameters for the Sentence-Transformer object as can be found at github.com/UKPLab/sentence-transformers/blob/master/sentence_transformers/SentenceTransformer.py. Thus, the optimizer was AdamW with a learning rate of $2 \cdot 10^{-5}$, weight decay was 0.01, and the learning rate schedular was set to WarmupLinear. We did not do a hyperparameter search, since that was not the point of our experiment.

[16] discuss.huggingface.co/t/train-the-best-sentence-embedding-model-ever-with-1b-training-pairs

Table I.1. Error rate (error), k-skewness (skew) and robinhood score (rh) on 20 Newsgroups (20), AG News (AG) and ten percent of Yahoo Answers (YA) for four pretrained models, base case vs after hubness reduction. Note that all these models have been trained on YA in various ways and their hubness on YA is in general lower than the models we trained. Thus, it seems that training on a specific dataset can make hubness go down for that dataset. However, hubness is still high on 20 which the models have not been trained on.

	20			**AG**			**YA**		
	error	**skew**	**rh**	**error**	**skew**	**rh**	**error**	**skew**	**rh**
multi-qa-distilbert-cos-v1									
base	0.318	7.58	0.37	0.081	1.90	0.59	0.271	2.12	0.69
f-norm	0.316	2.15	0.34	0.082	1.88	0.58	0.268	2.16	0.69
MP	0.292	13.06	0.28	0.081	1.55	0.55	0.271	1.77	0.66
f-norm + MP	0.289	1.89	0.30	0.079	1.57	0.56	0.269	1.88	0.67
all-MiniLM-L12-v2									
base	0.324	11.45	0.35	0.088	1.91	0.58	0.268	1.98	0.68
f-norm	0.324	2.22	0.33	0.086	1.88	0.58	0.269	1.97	0.68
MP	0.302	16.24	0.28	0.086	1.55	0.55	0.270	1.78	0.66
f-norm + MP	0.300	1.88	0.29	0.085	1.56	0.56	0.267	1.79	0.67
all-mpnet-base-v2									
base	0.295	11.53	0.37	0.082	1.81	0.58	0.251	2.02	0.68
f-norm	0.293	2.32	0.34	0.080	1.80	0.58	0.256	1.96	0.68
MP	0.272	15.80	0.30	0.085	1.57	0.55	0.250	1.73	0.66
f-norm + MP	0.272	1.98	0.30	0.082	1.54	0.56	0.250	1.78	0.67
all-distilroberta-v1									
base	0.322	6.61	0.39	0.085	1.93	0.59	0.254	2.08	0.68
f-norm	0.315	2.53	0.35	0.084	1.90	0.58	0.257	2.07	0.68
MP	0.291	11.20	0.29	0.087	1.57	0.55	0.254	1.72	0.66
f-norm + MP	0.296	2.05	0.31	0.085	1.59	0.56	0.253	1.82	0.67

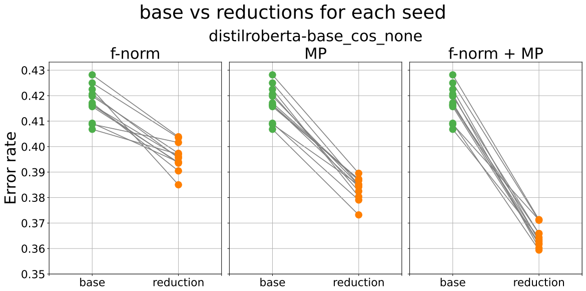


Figure G.2. 20 Newsgroups. Average base k-skewness 5.60.

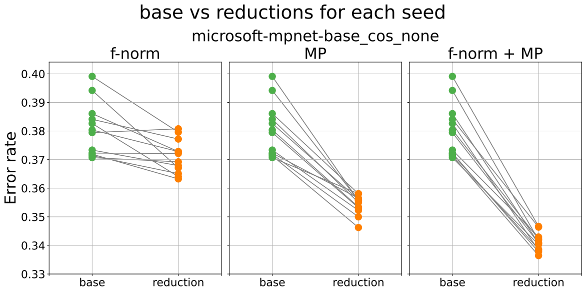


Figure G.3. 20 Newsgroups. Average base k-skewness 4.79.

J.2 GPU and CPU Types

Training of the models and making embeddings was done on the following GPU types: NVIDIA TITAN Xp, GeForce GTX 1080 Ti, NVIDIA TITAN V and NVIDIA TITAN X.

The knn and hubness measures were done on CPU on either AMD Ryzen 7 PRO 5850U or Xeon E5-2620 v4.

J.3 Computation Time

Training and generating embeddings times reported for running on an NVIDIA TITAN Xp GPU.

The total training time of the 540 models was about 28.5 hours. Of this, the models using microsoft-mpnet-base as base model took about 12 hours, the models using distilroberta-base as base model took about 6 hours and the models using microsoft-MiniLM-L12-H384-uncased as base model took about 10.5 hours. Note that a single model takes 4 minutes or less to train.

Total time generating embeddings for our own models was about 11.50 hours for 20 Newsgroups for all 540 models, about 21 hours for AG News for the 360 models using microsoft-mpnet-base or distilroberta-base as base model, and about 46.5 hours for the ten percent of Yahoo Answers for the 360 models using microsoft-mpnet-base or distilroberta-base as base model.

Generating embeddings for the four pretrained

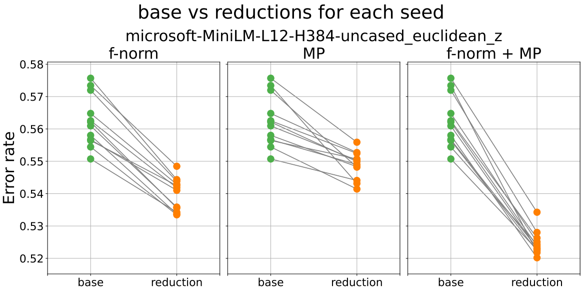


Figure G.4. 20 Newsgroups. Average base k-skewness 10.06.

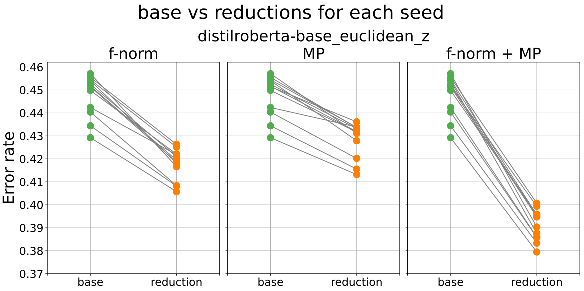


Figure G.5. 20 Newsgroups. Average base k-skewness 5.92.

models took about an hour. This gives us a total of 108.5 hours of GPU time for training and embeddings.

Times for knn and hubness measures reported for running on an AMD Ryzen 7 PRO 5850U CPU.

Getting knn results for our own models took about 69 hours for 20 Newsgroups for all 540 models, about 267 hours for AG News for the 360 models using microsoft-mpnet-base or distilroberta-base as base model, and about 306 hours for the ten percent of Yahoo Answers for the 360 models using microsoft-mpnet-base or distilroberta-base as base model.

Getting knn results from the pretrained models for all datasets took about 7 hours.

Getting the hubness results for our own models took about 7.5 hours for 20 Newsgroups for all 540 models, about 46.5 hours for AG News for the 360 models using microsoft-mpnet-base or distilroberta-base as base model, and about 60 hours for the ten percent of Yahoo Answers for the 360 models using microsoft-mpnet-base or distilroberta-base as base model.

Getting hubness results from the pretrained models for all datasts took about 1.5 hours.

Thus, in total we used 764.5 hours of CPU time for knn and hubness measures.

K Package Versions

Important packages: h5py 3.7.0, hdf5 1.10.6, nltk 3.7, numpy 1.21.5, python 3.9.7, pytorch 1.12.0, sci-kit-hubness 0.30.0a1, scikit-learn 1.1.1, sentence-transformers 2.2.2, transformers 4.20.1. Complete environment description can be found on github[17].

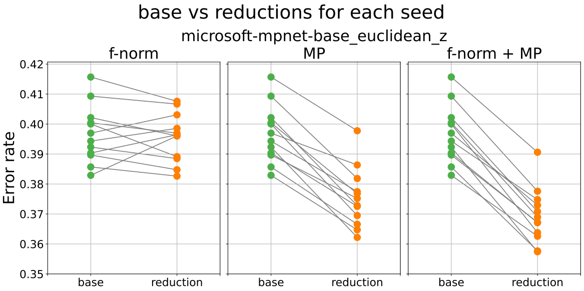


Figure G.6. 20 Newsgroups. Average base k-skewness 5.50.

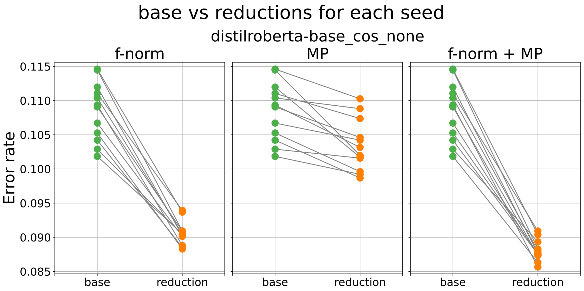


Figure G.7. AG News. Average base k-skewness 3.64.

L Training on STS benchmark

The STS benchmark dataset has pairs of sentences annotated with a semantic similarity score from 0 (not similar) to 5 (semantically equal). It is split into a train, a dev, and a test set with 7251, 1500, and 2886 sentence pairs, respectively. We trained our models on the train split. License: Unknown.

When training the models, we had to decide what distances these scores should correspond to. There were two natural choices: The maximum distance between embeddings should be the distance between orthogonal unit vectors, $\sqrt{2}$ when using Euclidean distance, 1 when using cosine distance. Or, the maximum distance between embeddings should be the distance between opposite unit vectors, 2 when using Euclidean distance or cosine distance. In some preliminary experiments we trained both orthogonal and opposite, but there did not seem to be much difference, so we went ahead using only orthogonal.

M Evaluation Details

When working on the large datasets, AG News and Yahoo Answers, we used the nmslib library [30] and the Hierarchical Navigable World graph (HNSW)

[17] https://github.com/bemigini/hubness-reduction-sentence-bert

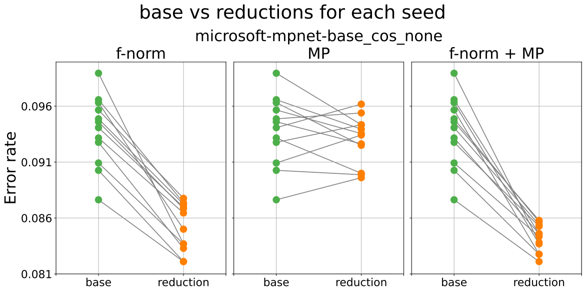


Figure G.8. AG News. Average base k-skewness 4.98.

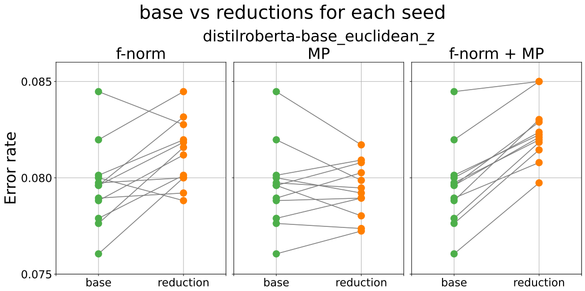


Figure G.9. AG News. Average base k-skewness 2.40.

method [31] for approximate nearest neighbours, which builds a slightly different index each time it is run.

When using f-norm at test time, for each dimension in the embeddings, we draw samples from a standard normal distribution for all training and test embeddings and replace values in the embeddings with the samples which preserve the ranking within the dimension. This procedure will also work if there is only one test point we want to classify.

N Best error rates

We consider which models with which hubness reduction methods give us the lowest error rate when taking the mean over the twelve random seeds.

Models using the largest base model (microsoft-mpnet-base) have the lowest mean error rate. If we take the three best error rates from the three datasets, they are all from models using the largest base model. On 20 Newsgroups using f-norm + MP gives the lowest mean error rate. On AG News, using Euclidean distance during training seemed the best predictor for a low error rate. On Yahoo Answers, using the largest base model, Euclidean distance and z-score normalisation during training gave the four lowest error rates with f-norm + MP being the lowest one. More details below.

On 20 Newsgroups, we find that the ten best mean error rates all used f-norm + MP. This is still true if we only look at models using the medium base model (distilroberta-base) or the small base model (microsoft-MiniLM-L12-H384-uncased). So on this dataset, using the f-norm + MP gives the best mean error rate.

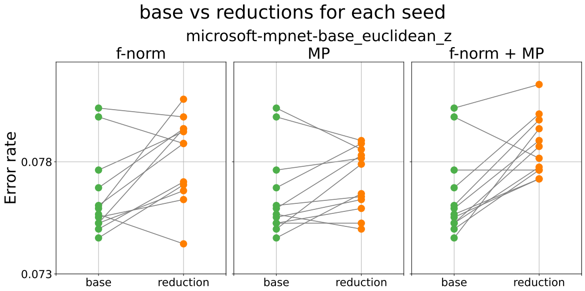


Figure G.10. AG News. Average base k-skewness 2.22.

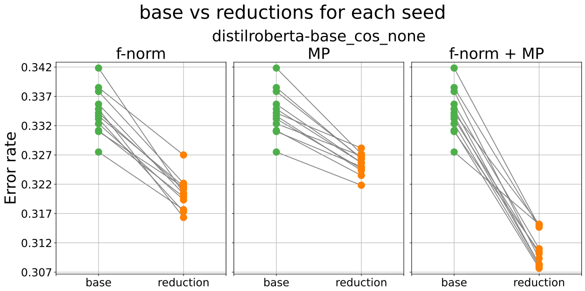


Figure G.11. Ten percent of Yahoo Answers. Average base k-skewness 8.41.

On AG News, the ten best mean error rates all used Euclidean distance for training. The best three also used z-score normalisation. The best three and the seventh best use the largest base model, the rest all use the medium base model. If we only look at models using the medium base model, the best ten still all used Euclidean distance for training, however the best four are no longer z-score normalisation. So on this dataset, using Euclidean distance for training the model seems to be the best predictor for a low mean error rate.

On the Yahoo Answers dataset, the best four all used the largest base model, Euclidean distance and z-score normalisation and with the reductions in order: f-norm + MP, MP, f-norm, none. The numbers four to ten all used the f-norm + MP. If we restrict to the medium base model, the best seven mean error rates are all from models which used Euclidean distance. The best three are after using f-norm + MP, four to six used f-norm, seven used MP and eight to ten used f-norm + MP. So on this dataset, f-norm + MP still gives the best mean error rate, but using the largest base model, Euclidean distance and z-score normalisation gave better error rate than using anything else and then f-norm + MP.

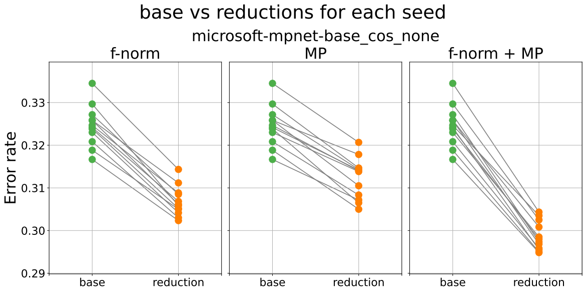


Figure G.12. Ten percent of Yahoo Answers. Average base k-skewness 5.54.

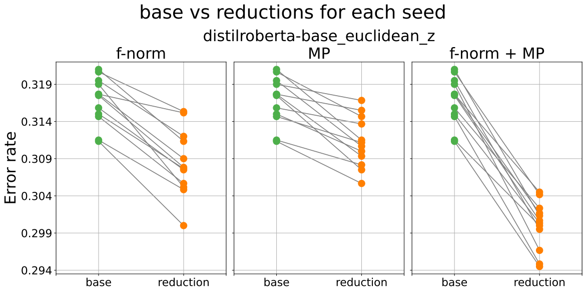


Figure G.13. Ten percent of Yahoo Answers. Average base k-skewness 3.29.

O Differences Between Normalisations During Training

Models which use some kind of embedding length normalisation (n; c,n; z) tend to have slightly lower hubness measured with robinhood score than models which do not have this kind of normalisation (none; c). However, there is also a difference between the base models used. On 20 Newsgroups, if we take any two models using microsoft-mpnet-base as base model where one uses some kind of length normalisation (n; c,n; z) and the other does not (none; c), then in about 97.2% of cases the model not using length normalisation will have the higher robinhood score. If we do the same for two models using microsoft-MiniLM-L12-H384-uncased as base model, the same will be true in about 70.5% cases. It is still a majority and it is still safe to assume it is a different distribution (Mann-Whitney U test gives a p-value of $3.3 \cdot 10^{-6}$), but it is not as overwhelming.

On 20 Newsgroups for the models using the small and medium base models, the base error rate is worse when using Euclidean distance without than it is with normalisation to unit length. However, for the models using the largest base model (except for one which seems to have failed training), error rates are slightly better for the models without normalisation than with. On AG News and Yahoo Answers, models (medium and large) using Euclidean distance and no normalisation have lower error rates than models with normalisation to unit length and also lower than models using cosine or cosine distance both with and without unit length normalisation. See Appendix H for figures.

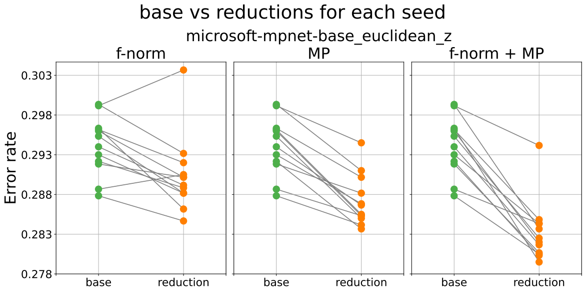


Figure G.14. Ten percent of Yahoo Answers. Average base k-skewness 3.05.

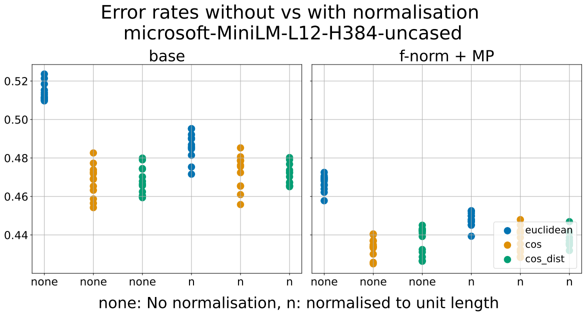


Figure H.1. 20 Newsgroups.

P Mutual Proximity Implementation

[12] showed that instead of calculating distances from x, y to all other points in the dataset, it is enough to take a sample of other points, and, if further assuming that distances are approximately normally distributed, stable results can be obtained with as few as 30 samples.

We used an implementation of mutual proximity which assumes a normal distribution of distances, since it is faster to compute than the one which allows arbitrary distribution of distances. This is reasonable, since [12] showed that mutual proximity assuming a normal distribution of distances, had a similar performance to the type without the assumption.

Q More Skewness Gives Larger Effect of Reduction

On AG News, for the 65 cases where error rate does not improve when using f-norm + MP, the maximum skewness is 2.58 and the mean skewness is 2.27, compared to 9.80 and 2.80 for models where error rate improved and 9.80 and 3.41 for models

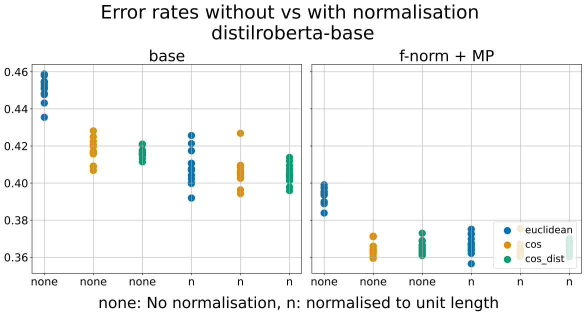


Figure H.2. 20 Newsgroups.

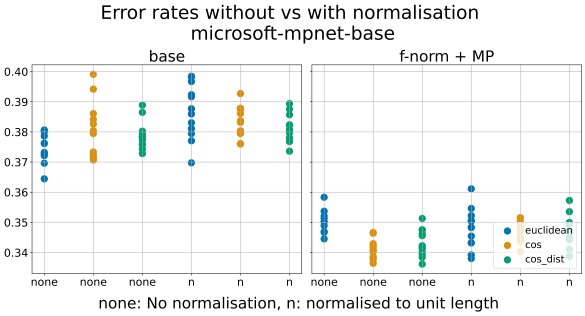


Figure H.3. 20 Newsgroups. One model (euclidean) with greatly deviating values (base error rate about 80%) has been removed to better show the variance in the remaining models.

where error rate was reduced with at least half a percentage point. There are 96 models which have a base skewness score of more than 3 and of these, 91 cases have an error rate reduction of more than half a percentage point. So the method is more likely to have a larger effect if the base skewness is higher.

R Note on SVM

Using SVM on our trained embeddings for classification on 20 Newsgroups, we saw that in 539 or 506 cases out of 540 when using an rbf or linear kernel, error rate improved from the base case when using f-norm on the embeddings before SVM. Error rate also went down in 455 or 540 cases out of 540 when using an rbf or linear kernel when using f-norm compared to Z-score normalisation before SVM. Exploring this further would be a good direction for future research.

S Overview Tables

Figures S.1, S.2, S.3, S.4, S.5, S.6, S.7 show tables with min, max and mean of error rate(error), k-skewness(skew) and robinhood score(rh) for the 12 random seeds of all our trained model types on the test splits of our datasets.

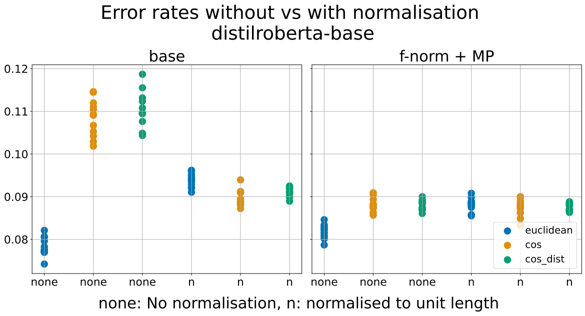


Figure H.4. AG News.

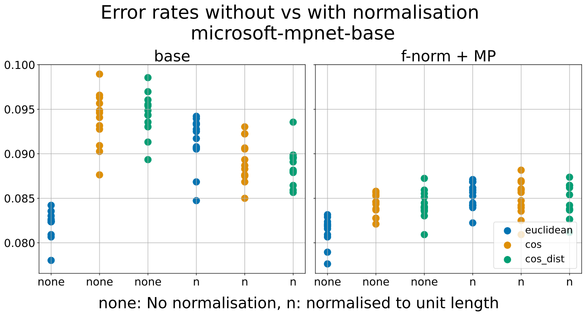


Figure H.5. AG News. One model (euclidean) with greatly deviating values (base error rate about 14%) has been removed to better show the variance in the remaining models.

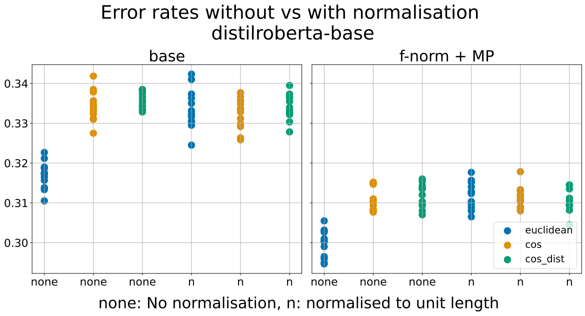


Figure H.6. Ten percent of Yahoo Answers.

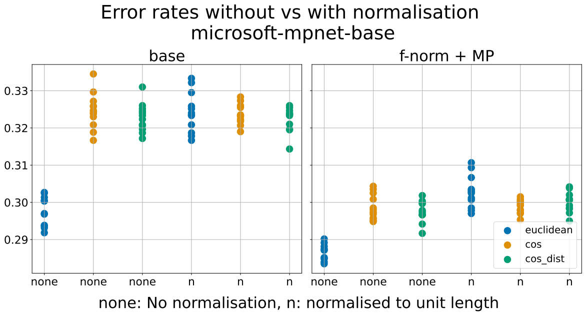
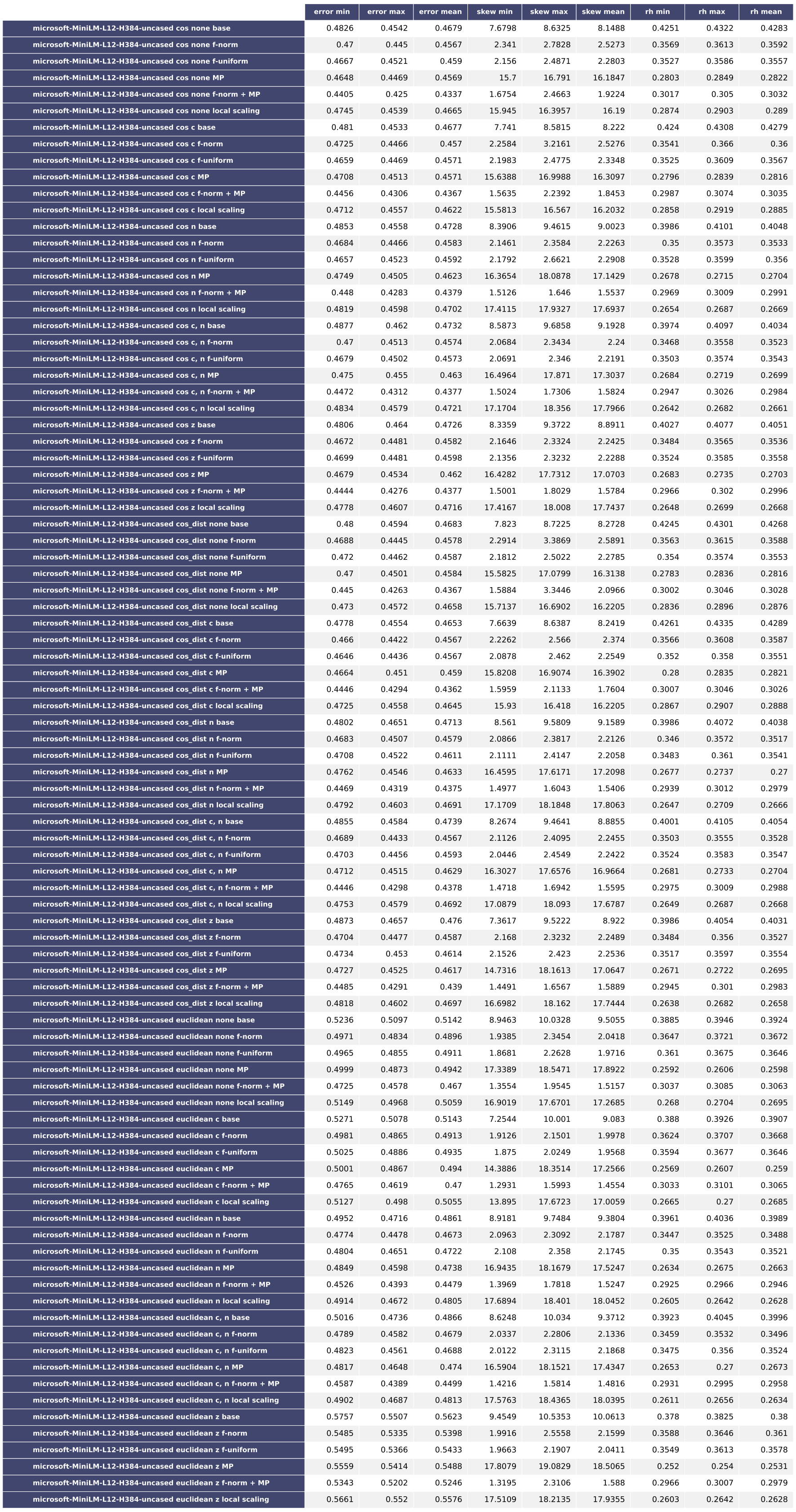
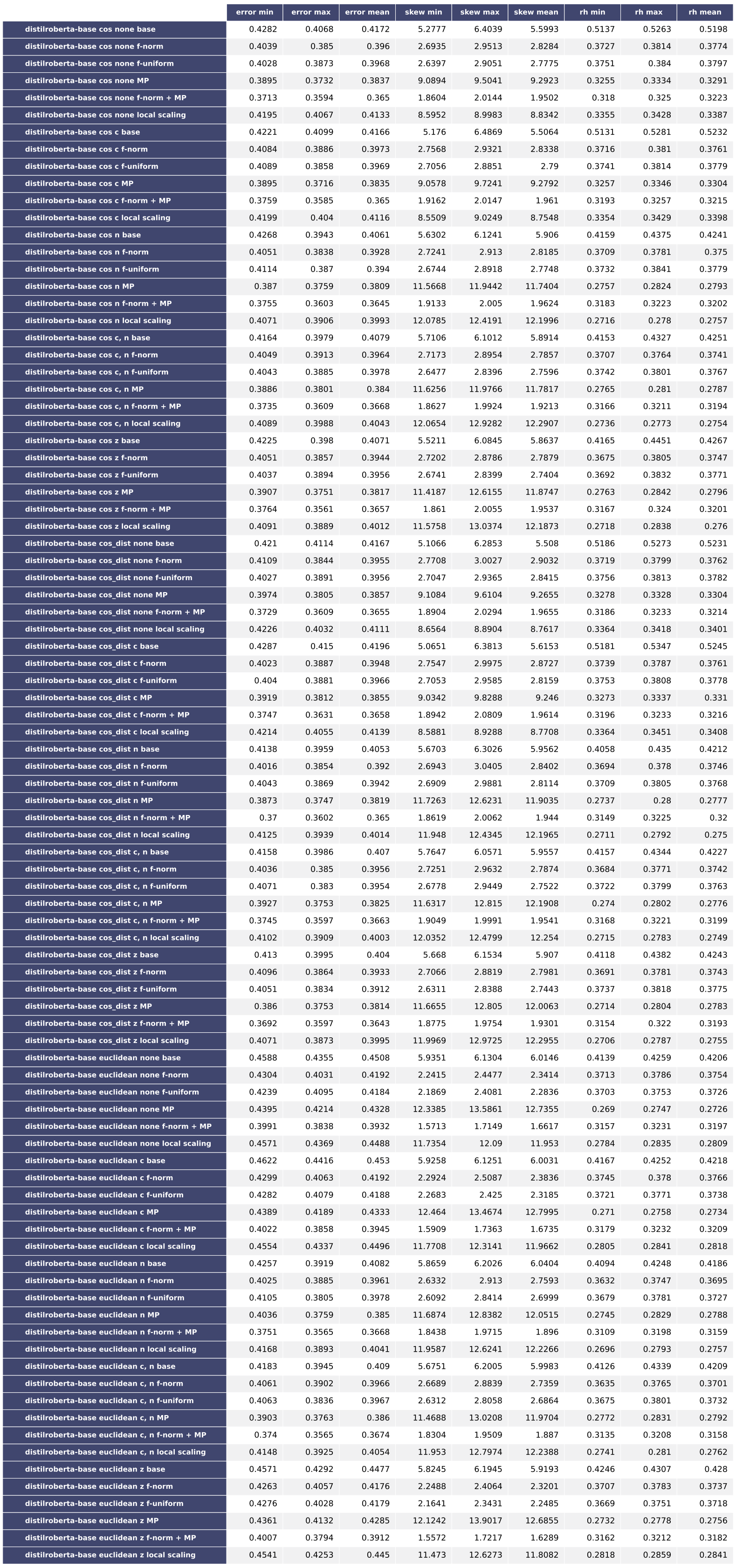


Figure H.7. Ten percent of Yahoo Answers. One model (euclidean) with greatly deviating values (base error rate about 55%) has been removed to better show the variance in the remaining models.



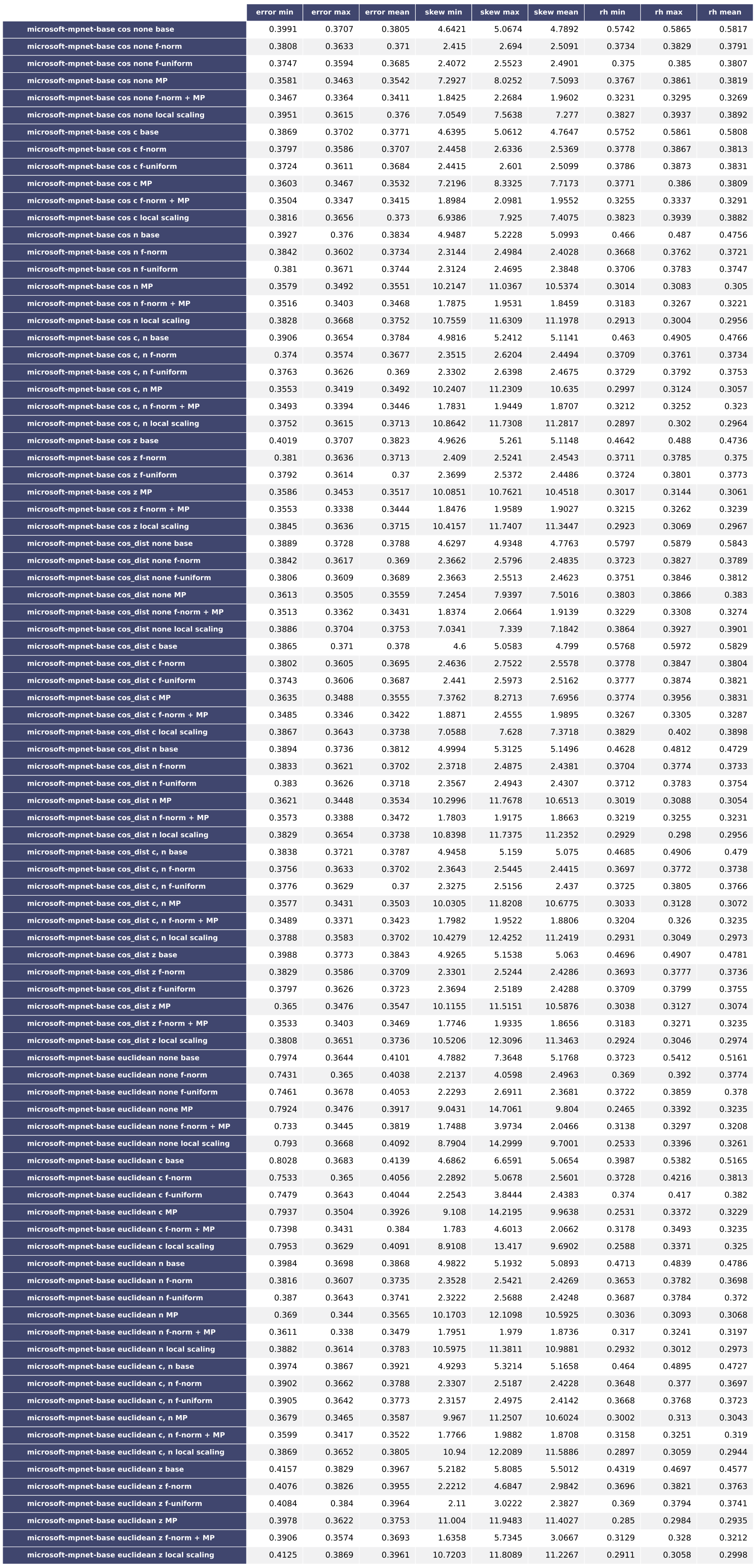
	error min	error max	error mean	skew min	skew max	skew mean	rh min	rh max	rh mean
microsoft-MiniLM-L12-H384-uncased cos none base	0.4826	0.4542	0.4679	7.6798	8.6325	8.1488	0.4251	0.4322	0.4283
microsoft-MiniLM-L12-H384-uncased cos none f-norm	0.47	0.445	0.4567	2.341	2.7828	2.5273	0.3569	0.3613	0.3592
microsoft-MiniLM-L12-H384-uncased cos none f-uniform	0.4667	0.4521	0.459	2.156	2.4871	2.2803	0.3527	0.3586	0.3557
microsoft-MiniLM-L12-H384-uncased cos none MP	0.4648	0.4469	0.4569	15.7	16.791	16.1847	0.2803	0.2849	0.2822
microsoft-MiniLM-L12-H384-uncased cos none f-norm + MP	0.4405	0.425	0.4337	1.6754	2.4663	1.9224	0.3017	0.305	0.3032
microsoft-MiniLM-L12-H384-uncased cos none local scaling	0.4745	0.4539	0.4665	15.945	16.3957	16.19	0.2874	0.2903	0.289
microsoft-MiniLM-L12-H384-uncased cos c base	0.481	0.4533	0.4677	7.741	8.5815	8.222	0.424	0.4308	0.4279
microsoft-MiniLM-L12-H384-uncased cos c f-norm	0.4725	0.4466	0.457	2.2584	3.2161	2.5276	0.3541	0.366	0.36
microsoft-MiniLM-L12-H384-uncased cos c f-uniform	0.4659	0.4469	0.4571	2.1983	2.4775	2.3348	0.3525	0.3609	0.3567
microsoft-MiniLM-L12-H384-uncased cos c MP	0.4708	0.4513	0.4571	15.6388	16.9988	16.3097	0.2796	0.2839	0.2816
microsoft-MiniLM-L12-H384-uncased cos c f-norm + MP	0.4456	0.4306	0.4367	1.5635	2.2392	1.8453	0.2987	0.3074	0.3035
microsoft-MiniLM-L12-H384-uncased cos c local scaling	0.4712	0.4557	0.4622	15.5813	16.567	16.2032	0.2858	0.2919	0.2885
microsoft-MiniLM-L12-H384-uncased cos n base	0.4853	0.4558	0.4728	8.3906	9.4615	9.0023	0.3986	0.4101	0.4048
microsoft-MiniLM-L12-H384-uncased cos n f-norm	0.4684	0.4466	0.4583	2.1461	2.3584	2.2263	0.35	0.3573	0.3533
microsoft-MiniLM-L12-H384-uncased cos n f-uniform	0.4657	0.4523	0.4592	2.1792	2.6621	2.2908	0.3528	0.3599	0.356
microsoft-MiniLM-L12-H384-uncased cos n MP	0.4749	0.4505	0.4623	16.3654	18.0878	17.1429	0.2678	0.2715	0.2704
microsoft-MiniLM-L12-H384-uncased cos n f-norm + MP	0.448	0.4283	0.4379	1.5126	1.646	1.5537	0.2969	0.3009	0.2991
microsoft-MiniLM-L12-H384-uncased cos n local scaling	0.4819	0.4598	0.4702	17.4115	17.9327	17.6937	0.2654	0.2687	0.2669
microsoft-MiniLM-L12-H384-uncased cos c, n base	0.4877	0.462	0.4732	8.5873	9.6858	9.1928	0.3974	0.4097	0.4034
microsoft-MiniLM-L12-H384-uncased cos c, n f-norm	0.47	0.4513	0.4574	2.0684	2.3434	2.24	0.3468	0.3558	0.3523
microsoft-MiniLM-L12-H384-uncased cos c, n f-uniform	0.4679	0.4502	0.4573	2.0691	2.346	2.2191	0.3503	0.3574	0.3543
microsoft-MiniLM-L12-H384-uncased cos c, n MP	0.475	0.455	0.463	16.4964	17.871	17.3037	0.2684	0.2719	0.2699
microsoft-MiniLM-L12-H384-uncased cos c, n f-norm + MP	0.4472	0.4312	0.4377	1.5024	1.7306	1.5824	0.2947	0.3026	0.2984
microsoft-MiniLM-L12-H384-uncased cos c, n local scaling	0.4834	0.4579	0.4721	17.1704	18.356	17.7966	0.2642	0.2682	0.2661
microsoft-MiniLM-L12-H384-uncased cos z base	0.4806	0.464	0.4726	8.3359	9.3722	8.8911	0.4027	0.4077	0.4051
microsoft-MiniLM-L12-H384-uncased cos z f-norm	0.4672	0.4481	0.4582	2.1646	2.3324	2.2425	0.3484	0.3565	0.3536
microsoft-MiniLM-L12-H384-uncased cos z f-uniform	0.4699	0.4481	0.4598	2.1356	2.3232	2.2288	0.3524	0.3585	0.3558
microsoft-MiniLM-L12-H384-uncased cos z MP	0.4679	0.4534	0.462	16.4282	17.7312	17.0703	0.2683	0.2735	0.2703
microsoft-MiniLM-L12-H384-uncased cos z f-norm + MP	0.4444	0.4276	0.4377	1.5001	1.8029	1.5784	0.2966	0.302	0.2996
microsoft-MiniLM-L12-H384-uncased cos z local scaling	0.4778	0.4607	0.4716	17.4167	18.008	17.7437	0.2648	0.2699	0.2668
microsoft-MiniLM-L12-H384-uncased cos_dist none base	0.48	0.4594	0.4683	7.823	8.7225	8.2728	0.4245	0.4301	0.4268
microsoft-MiniLM-L12-H384-uncased cos_dist none f-norm	0.4688	0.4445	0.4578	2.2914	3.3869	2.5891	0.3563	0.3615	0.3588
microsoft-MiniLM-L12-H384-uncased cos_dist none f-uniform	0.472	0.4462	0.4587	2.1812	2.5022	2.2785	0.354	0.3574	0.3553
microsoft-MiniLM-L12-H384-uncased cos_dist none MP	0.47	0.4501	0.4584	15.5825	17.0799	16.3138	0.2783	0.2836	0.2816
microsoft-MiniLM-L12-H384-uncased cos_dist none f-norm + MP	0.445	0.4263	0.4367	1.5884	3.3446	2.0966	0.3002	0.3046	0.3028
microsoft-MiniLM-L12-H384-uncased cos_dist none local scaling	0.473	0.4572	0.4658	15.7137	16.6902	16.2205	0.2836	0.2896	0.2876
microsoft-MiniLM-L12-H384-uncased cos_dist c base	0.4778	0.4554	0.4653	7.6639	8.6387	8.2419	0.4261	0.4335	0.4289
microsoft-MiniLM-L12-H384-uncased cos_dist c f-norm	0.466	0.4422	0.4567	2.2262	2.566	2.374	0.3566	0.3608	0.3587
microsoft-MiniLM-L12-H384-uncased cos_dist c f-uniform	0.4646	0.4436	0.4567	2.0878	2.462	2.2549	0.352	0.358	0.3551
microsoft-MiniLM-L12-H384-uncased cos_dist c MP	0.4664	0.451	0.459	15.8208	16.9074	16.3902	0.28	0.2835	0.2821
microsoft-MiniLM-L12-H384-uncased cos_dist c f-norm + MP	0.4446	0.4294	0.4362	1.5959	2.1133	1.7604	0.3007	0.3046	0.3026
microsoft-MiniLM-L12-H384-uncased cos_dist c local scaling	0.4725	0.4558	0.4645	15.93	16.418	16.2205	0.2867	0.2907	0.2888
microsoft-MiniLM-L12-H384-uncased cos_dist n base	0.4802	0.4651	0.4713	8.561	9.5809	9.1589	0.3986	0.4072	0.4038
microsoft-MiniLM-L12-H384-uncased cos_dist n f-norm	0.4683	0.4507	0.4579	2.0866	2.3817	2.2126	0.346	0.3572	0.3517
microsoft-MiniLM-L12-H384-uncased cos_dist n f-uniform	0.4708	0.4522	0.4611	2.1111	2.4147	2.2058	0.3483	0.361	0.3541
microsoft-MiniLM-L12-H384-uncased cos_dist n MP	0.4762	0.4546	0.4633	16.4595	17.6171	17.2098	0.2677	0.2737	0.27
microsoft-MiniLM-L12-H384-uncased cos_dist n f-norm + MP	0.4469	0.4319	0.4375	1.4977	1.6043	1.5406	0.2939	0.3012	0.2979
microsoft-MiniLM-L12-H384-uncased cos_dist n local scaling	0.4792	0.4603	0.4691	17.1709	18.1848	17.8063	0.2647	0.2709	0.2666
microsoft-MiniLM-L12-H384-uncased cos_dist c, n base	0.4855	0.4584	0.4739	8.2674	9.4641	8.8855	0.4001	0.4105	0.4054
microsoft-MiniLM-L12-H384-uncased cos_dist c, n f-norm	0.4689	0.4433	0.4567	2.1126	2.4095	2.2455	0.3503	0.3555	0.3528
microsoft-MiniLM-L12-H384-uncased cos_dist c, n f-uniform	0.4703	0.4456	0.4593	2.0446	2.4549	2.2422	0.3524	0.3583	0.3547
microsoft-MiniLM-L12-H384-uncased cos_dist c, n MP	0.4712	0.4515	0.4629	16.3027	17.6576	16.9664	0.2681	0.2733	0.2704
microsoft-MiniLM-L12-H384-uncased cos_dist c, n f-norm + MP	0.4446	0.4298	0.4378	1.4718	1.6942	1.5595	0.2975	0.3009	0.2988
microsoft-MiniLM-L12-H384-uncased cos_dist c, n local scaling	0.4753	0.4579	0.4692	17.0879	18.093	17.6787	0.2649	0.2687	0.2668
microsoft-MiniLM-L12-H384-uncased cos_dist z base	0.4873	0.4657	0.476	7.3617	9.5222	8.922	0.3986	0.4054	0.4031
microsoft-MiniLM-L12-H384-uncased cos_dist z f-norm	0.4704	0.4477	0.4587	2.168	2.3232	2.2489	0.3484	0.356	0.3527
microsoft-MiniLM-L12-H384-uncased cos_dist z f-uniform	0.4734	0.453	0.4614	2.1526	2.423	2.2536	0.3517	0.3597	0.3554
microsoft-MiniLM-L12-H384-uncased cos_dist z MP	0.4727	0.4525	0.4617	14.7316	18.1613	17.0647	0.2671	0.2722	0.2695
microsoft-MiniLM-L12-H384-uncased cos_dist z f-norm + MP	0.4485	0.4291	0.439	1.4491	1.6567	1.5889	0.2945	0.301	0.2983
microsoft-MiniLM-L12-H384-uncased cos_dist z local scaling	0.4818	0.4602	0.4697	16.6982	18.162	17.7444	0.2638	0.2682	0.2658
microsoft-MiniLM-L12-H384-uncased euclidean none base	0.5236	0.5097	0.5142	8.9463	10.0328	9.5055	0.3885	0.3946	0.3924
microsoft-MiniLM-L12-H384-uncased euclidean none f-norm	0.4971	0.4834	0.4896	1.9385	2.3454	2.0418	0.3647	0.3721	0.3672
microsoft-MiniLM-L12-H384-uncased euclidean none f-uniform	0.4965	0.4855	0.4911	1.8681	2.2628	1.9716	0.361	0.3675	0.3646
microsoft-MiniLM-L12-H384-uncased euclidean none MP	0.4999	0.4873	0.4942	17.3389	18.5471	17.8922	0.2592	0.2606	0.2598
microsoft-MiniLM-L12-H384-uncased euclidean none f-norm + MP	0.4725	0.4578	0.467	1.3554	1.9545	1.5157	0.3037	0.3085	0.3063
microsoft-MiniLM-L12-H384-uncased euclidean none local scaling	0.5149	0.4968	0.5059	16.9019	17.6701	17.2685	0.268	0.2704	0.2695
microsoft-MiniLM-L12-H384-uncased euclidean c base	0.5271	0.5078	0.5143	7.2544	10.001	9.083	0.388	0.3926	0.3907
microsoft-MiniLM-L12-H384-uncased euclidean c f-norm	0.4981	0.4865	0.4913	1.9126	2.1501	1.9978	0.3624	0.3707	0.3668
microsoft-MiniLM-L12-H384-uncased euclidean c f-uniform	0.5025	0.4886	0.4935	1.875	2.0249	1.9568	0.3594	0.3677	0.3646
microsoft-MiniLM-L12-H384-uncased euclidean c MP	0.5001	0.4867	0.494	14.3886	18.3514	17.2566	0.2569	0.2607	0.259
microsoft-MiniLM-L12-H384-uncased euclidean c f-norm + MP	0.4765	0.4619	0.47	1.2931	1.5993	1.4554	0.3033	0.3101	0.3065
microsoft-MiniLM-L12-H384-uncased euclidean c local scaling	0.5127	0.498	0.5055	13.895	17.6723	17.0059	0.2665	0.27	0.2685
microsoft-MiniLM-L12-H384-uncased euclidean n base	0.4952	0.4716	0.4861	8.9181	9.7484	9.3804	0.3961	0.4036	0.3989
microsoft-MiniLM-L12-H384-uncased euclidean n f-norm	0.4774	0.4478	0.4673	2.0963	2.3092	2.1787	0.3447	0.3525	0.3488
microsoft-MiniLM-L12-H384-uncased euclidean n f-uniform	0.4804	0.4651	0.4722	2.108	2.358	2.1745	0.35	0.3543	0.3521
microsoft-MiniLM-L12-H384-uncased euclidean n MP	0.4849	0.4598	0.4738	16.9435	18.1679	17.5247	0.2634	0.2675	0.2663
microsoft-MiniLM-L12-H384-uncased euclidean n f-norm + MP	0.4526	0.4393	0.4479	1.3969	1.7818	1.5247	0.2925	0.2966	0.2946
microsoft-MiniLM-L12-H384-uncased euclidean n local scaling	0.4914	0.4672	0.4805	17.6894	18.401	18.0452	0.2605	0.2642	0.2628
microsoft-MiniLM-L12-H384-uncased euclidean c, n base	0.5016	0.4736	0.4866	8.6248	10.034	9.3712	0.3923	0.4045	0.3996
microsoft-MiniLM-L12-H384-uncased euclidean c, n f-norm	0.4789	0.4582	0.4679	2.0337	2.2806	2.1336	0.3459	0.3532	0.3496
microsoft-MiniLM-L12-H384-uncased euclidean c, n f-uniform	0.4823	0.4561	0.4688	2.0122	2.3115	2.1868	0.3475	0.356	0.3524
microsoft-MiniLM-L12-H384-uncased euclidean c, n MP	0.4817	0.4648	0.474	16.5904	18.1521	17.4347	0.2653	0.27	0.2673
microsoft-MiniLM-L12-H384-uncased euclidean c, n f-norm + MP	0.4587	0.4389	0.4499	1.4216	1.5814	1.4816	0.2931	0.2995	0.2958
microsoft-MiniLM-L12-H384-uncased euclidean c, n local scaling	0.4902	0.4687	0.4813	17.5763	18.4365	18.0395	0.2611	0.2656	0.2634
microsoft-MiniLM-L12-H384-uncased euclidean z base	0.5757	0.5507	0.5623	9.4549	10.5353	10.0613	0.378	0.3825	0.38
microsoft-MiniLM-L12-H384-uncased euclidean z f-norm	0.5485	0.5335	0.5398	1.9916	2.5558	2.1599	0.3588	0.3646	0.361
microsoft-MiniLM-L12-H384-uncased euclidean z f-uniform	0.5495	0.5366	0.5433	1.9663	2.1907	2.0411	0.3549	0.3613	0.3578
microsoft-MiniLM-L12-H384-uncased euclidean z MP	0.5559	0.5414	0.5488	17.8079	19.0829	18.5065	0.252	0.254	0.2531
microsoft-MiniLM-L12-H384-uncased euclidean z f-norm + MP	0.5343	0.5202	0.5246	1.3195	2.3106	1.588	0.2966	0.3007	0.2979
microsoft-MiniLM-L12-H384-uncased euclidean z local scaling	0.5661	0.552	0.5576	17.5109	18.2135	17.9355	0.2603	0.2642	0.2628

Figure S.1. Error rate and hubness on the test split of the 20 Newsgroups dataset for the smallest base model tested. Means are over the 12 random seeds. error: Error rate, skew: K-skewness, rh: Robinhood score.



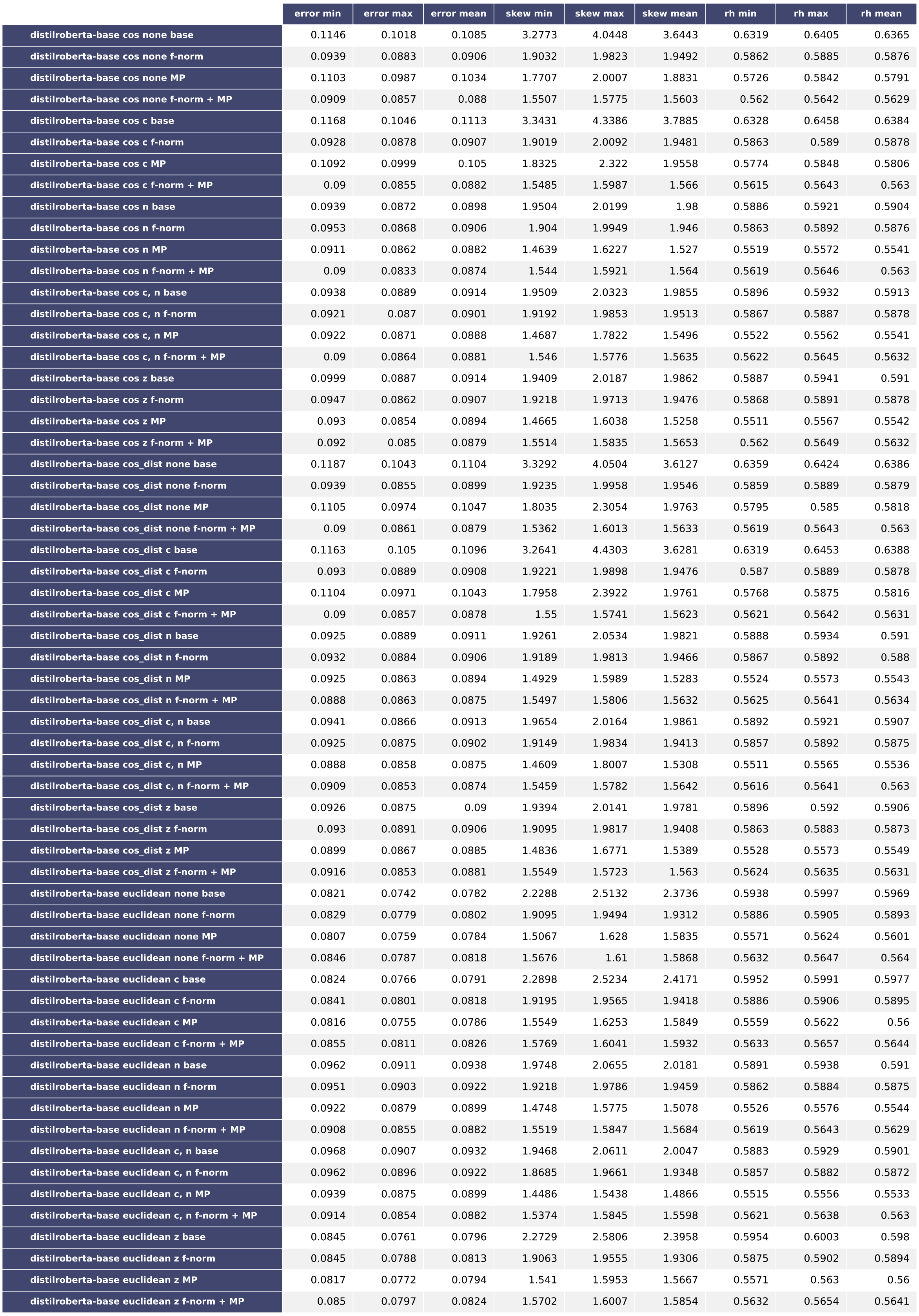
	error min	error max	error mean	skew min	skew max	skew mean	rh min	rh max	rh mean
distilroberta-base cos none base	0.4282	0.4068	0.4172	5.2777	6.4039	5.5993	0.5137	0.5263	0.5198
distilroberta-base cos none f-norm	0.4039	0.385	0.396	2.6935	2.9513	2.8284	0.3727	0.3814	0.3774
distilroberta-base cos none f-uniform	0.4028	0.3873	0.3968	2.6397	2.9051	2.7775	0.3751	0.384	0.3797
distilroberta-base cos none MP	0.3895	0.3732	0.3837	9.0894	9.5041	9.2923	0.3255	0.3334	0.3291
distilroberta-base cos none f-norm + MP	0.3713	0.3594	0.365	1.8604	2.0144	1.9502	0.318	0.325	0.3223
distilroberta-base cos none local scaling	0.4195	0.4067	0.4133	8.5952	8.9983	8.8342	0.3355	0.3428	0.3387
distilroberta-base cos c base	0.4221	0.4099	0.4166	5.176	6.4869	5.5064	0.5131	0.5281	0.5232
distilroberta-base cos c f-norm	0.4084	0.3886	0.3973	2.7568	2.9321	2.8338	0.3716	0.381	0.3761
distilroberta-base cos c f-uniform	0.4089	0.3858	0.3969	2.7056	2.8851	2.79	0.3741	0.3814	0.3779
distilroberta-base cos c MP	0.3895	0.3716	0.3835	9.0578	9.7241	9.2792	0.3257	0.3346	0.3304
distilroberta-base cos c f-norm + MP	0.3759	0.3585	0.365	1.9162	2.0147	1.961	0.3193	0.3257	0.3215
distilroberta-base cos c local scaling	0.4199	0.404	0.4116	8.5509	9.0249	8.7548	0.3354	0.3429	0.3398
distilroberta-base cos n base	0.4268	0.3943	0.4061	5.6302	6.1241	5.906	0.4159	0.4375	0.4241
distilroberta-base cos n f-norm	0.4051	0.3838	0.3928	2.7241	2.913	2.8185	0.3709	0.3781	0.375
distilroberta-base cos n f-uniform	0.4114	0.387	0.394	2.6744	2.8918	2.7748	0.3732	0.3841	0.3779
distilroberta-base cos n MP	0.387	0.3759	0.3809	11.5668	11.9442	11.7404	0.2757	0.2824	0.2793
distilroberta-base cos n f-norm + MP	0.3755	0.3603	0.3645	1.9133	2.005	1.9624	0.3183	0.3223	0.3202
distilroberta-base cos n local scaling	0.4071	0.3906	0.3993	12.0785	12.4191	12.1996	0.2716	0.278	0.2757
distilroberta-base cos c, n base	0.4164	0.3979	0.4079	5.7106	6.1012	5.8914	0.4153	0.4327	0.4251
distilroberta-base cos c, n f-norm	0.4049	0.3913	0.3964	2.7173	2.8954	2.7857	0.3707	0.3764	0.3741
distilroberta-base cos c, n f-uniform	0.4043	0.3885	0.3978	2.6477	2.8396	2.7596	0.3742	0.3801	0.3767
distilroberta-base cos c, n MP	0.3886	0.3801	0.384	11.6256	11.9766	11.7817	0.2765	0.281	0.2787
distilroberta-base cos c, n f-norm + MP	0.3735	0.3609	0.3668	1.8627	1.9924	1.9213	0.3166	0.3211	0.3194
distilroberta-base cos c, n local scaling	0.4089	0.3988	0.4043	12.0654	12.9282	12.2907	0.2736	0.2773	0.2754
distilroberta-base cos z base	0.4225	0.398	0.4071	5.5211	6.0845	5.8637	0.4165	0.4451	0.4267
distilroberta-base cos z f-norm	0.4051	0.3857	0.3944	2.7202	2.8786	2.7879	0.3675	0.3805	0.3747
distilroberta-base cos z f-uniform	0.4037	0.3894	0.3956	2.6741	2.8399	2.7404	0.3692	0.3832	0.3771
distilroberta-base cos z MP	0.3907	0.3751	0.3817	11.4187	12.6155	11.8747	0.2763	0.2842	0.2796
distilroberta-base cos z f-norm + MP	0.3764	0.3561	0.3657	1.861	2.0055	1.9537	0.3167	0.324	0.3201
distilroberta-base cos z local scaling	0.4091	0.3889	0.4012	11.5758	13.0374	12.1873	0.2718	0.2838	0.276
distilroberta-base cos_dist none base	0.421	0.4114	0.4167	5.1066	6.2853	5.508	0.5186	0.5273	0.5231
distilroberta-base cos_dist none f-norm	0.4109	0.3844	0.3955	2.7708	3.0027	2.9032	0.3719	0.3799	0.3762
distilroberta-base cos_dist none f-uniform	0.4027	0.3891	0.3956	2.7047	2.9365	2.8415	0.3756	0.3813	0.3782
distilroberta-base cos_dist none MP	0.3974	0.3805	0.3857	9.1084	9.6104	9.2655	0.3278	0.3328	0.3304
distilroberta-base cos_dist none f-norm + MP	0.3729	0.3609	0.3655	1.8904	2.0294	1.9655	0.3186	0.3233	0.3214
distilroberta-base cos_dist none local scaling	0.4226	0.4032	0.4111	8.6564	8.8904	8.7617	0.3364	0.3418	0.3401
distilroberta-base cos_dist c base	0.4287	0.415	0.4196	5.0651	6.3813	5.6153	0.5181	0.5347	0.5245
distilroberta-base cos_dist c f-norm	0.4023	0.3887	0.3948	2.7547	2.9975	2.8727	0.3739	0.3787	0.3761
distilroberta-base cos_dist c f-uniform	0.404	0.3881	0.3966	2.7053	2.9585	2.8159	0.3753	0.3808	0.3778
distilroberta-base cos_dist c MP	0.3919	0.3812	0.3855	9.0342	9.8288	9.246	0.3273	0.3337	0.331
distilroberta-base cos_dist c f-norm + MP	0.3747	0.3631	0.3658	1.8942	2.0809	1.9614	0.3196	0.3233	0.3216
distilroberta-base cos_dist c local scaling	0.4214	0.4055	0.4139	8.5881	8.9288	8.7708	0.3364	0.3451	0.3408
distilroberta-base cos_dist n base	0.4138	0.3959	0.4053	5.6703	6.3026	5.9562	0.4058	0.435	0.4212
distilroberta-base cos_dist n f-norm	0.4016	0.3854	0.392	2.6943	3.0405	2.8402	0.3694	0.378	0.3746
distilroberta-base cos_dist n f-uniform	0.4043	0.3869	0.3942	2.6909	2.9881	2.8114	0.3709	0.3805	0.3768
distilroberta-base cos_dist n MP	0.3873	0.3747	0.3819	11.7263	12.6231	11.9035	0.2737	0.28	0.2777
distilroberta-base cos_dist n f-norm + MP	0.37	0.3602	0.365	1.8619	2.0062	1.944	0.3149	0.3225	0.32
distilroberta-base cos_dist n local scaling	0.4125	0.3939	0.4014	11.948	12.4345	12.1965	0.2711	0.2792	0.275
distilroberta-base cos_dist c, n base	0.4158	0.3986	0.407	5.7647	6.0571	5.9557	0.4157	0.4344	0.4227
distilroberta-base cos_dist c, n f-norm	0.4036	0.385	0.3956	2.7251	2.9632	2.7874	0.3684	0.3771	0.3742
distilroberta-base cos_dist c, n f-uniform	0.4071	0.383	0.3954	2.6778	2.9449	2.7522	0.3722	0.3799	0.3763
distilroberta-base cos_dist c, n MP	0.3927	0.3753	0.3825	11.6317	12.815	12.1908	0.274	0.2802	0.2776
distilroberta-base cos_dist c, n f-norm + MP	0.3745	0.3597	0.3663	1.9049	1.9991	1.9541	0.3168	0.3221	0.3199
distilroberta-base cos_dist c, n local scaling	0.4102	0.3909	0.4003	12.0352	12.4799	12.254	0.2715	0.2783	0.2749
distilroberta-base cos_dist z base	0.413	0.3995	0.404	5.668	6.1534	5.907	0.4118	0.4382	0.4243
distilroberta-base cos_dist z f-norm	0.4096	0.3864	0.3933	2.7066	2.8819	2.7981	0.3691	0.3781	0.3743
distilroberta-base cos_dist z f-uniform	0.4051	0.3834	0.3912	2.6311	2.8388	2.7443	0.3737	0.3818	0.3775
distilroberta-base cos_dist z MP	0.386	0.3753	0.3814	11.6655	12.805	12.0063	0.2714	0.2804	0.2783
distilroberta-base cos_dist z f-norm + MP	0.3692	0.3597	0.3643	1.8775	1.9754	1.9301	0.3154	0.322	0.3193
distilroberta-base cos_dist z local scaling	0.4071	0.3873	0.3995	11.9969	12.9725	12.2955	0.2706	0.2787	0.2755
distilroberta-base euclidean none base	0.4588	0.4355	0.4508	5.9351	6.1304	6.0146	0.4139	0.4259	0.4206
distilroberta-base euclidean none f-norm	0.4304	0.4031	0.4192	2.2415	2.4477	2.3414	0.3713	0.3786	0.3754
distilroberta-base euclidean none f-uniform	0.4239	0.4095	0.4184	2.1869	2.4081	2.2836	0.3703	0.3753	0.3726
distilroberta-base euclidean none MP	0.4395	0.4214	0.4328	12.3385	13.5861	12.7355	0.269	0.2747	0.2726
distilroberta-base euclidean none f-norm + MP	0.3991	0.3838	0.3932	1.5713	1.7149	1.6617	0.3157	0.3231	0.3197
distilroberta-base euclidean none local scaling	0.4571	0.4369	0.4488	11.7354	12.09	11.953	0.2784	0.2835	0.2809
distilroberta-base euclidean c base	0.4622	0.4416	0.453	5.9258	6.1251	6.0031	0.4167	0.4252	0.4218
distilroberta-base euclidean c f-norm	0.4299	0.4063	0.4192	2.2924	2.5087	2.3836	0.3745	0.378	0.3766
distilroberta-base euclidean c f-uniform	0.4282	0.4079	0.4188	2.2683	2.425	2.3185	0.3721	0.3771	0.3738
distilroberta-base euclidean c MP	0.4389	0.4189	0.4333	12.464	13.4674	12.7995	0.271	0.2758	0.2734
distilroberta-base euclidean c f-norm + MP	0.4022	0.3858	0.3945	1.5909	1.7363	1.6735	0.3179	0.3232	0.3209
distilroberta-base euclidean c local scaling	0.4554	0.4337	0.4496	11.7708	12.3141	11.9662	0.2805	0.2841	0.2818
distilroberta-base euclidean n base	0.4257	0.3919	0.4082	5.8659	6.2026	6.0404	0.4094	0.4248	0.4186
distilroberta-base euclidean n f-norm	0.4025	0.3885	0.3961	2.6332	2.913	2.7593	0.3632	0.3747	0.3695
distilroberta-base euclidean n f-uniform	0.4105	0.3805	0.3978	2.6092	2.8414	2.6999	0.3679	0.3781	0.3727
distilroberta-base euclidean n MP	0.4036	0.3759	0.385	11.6874	12.8382	12.0515	0.2745	0.2829	0.2788
distilroberta-base euclidean n f-norm + MP	0.3751	0.3565	0.3668	1.8438	1.9715	1.896	0.3109	0.3198	0.3159
distilroberta-base euclidean n local scaling	0.4168	0.3893	0.4041	11.9587	12.6241	12.2266	0.2696	0.2793	0.2757
distilroberta-base euclidean c, n base	0.4183	0.3945	0.409	5.6751	6.2005	5.9983	0.4126	0.4339	0.4209
distilroberta-base euclidean c, n f-norm	0.4061	0.3902	0.3966	2.6689	2.8839	2.7359	0.3635	0.3765	0.3701
distilroberta-base euclidean c, n f-uniform	0.4063	0.3836	0.3967	2.6312	2.8058	2.6864	0.3675	0.3801	0.3732
distilroberta-base euclidean c, n MP	0.3903	0.3763	0.386	11.4688	13.0208	11.9704	0.2772	0.2831	0.2792
distilroberta-base euclidean c, n f-norm + MP	0.374	0.3565	0.3674	1.8304	1.9509	1.887	0.3135	0.3208	0.3158
distilroberta-base euclidean c, n local scaling	0.4148	0.3925	0.4054	11.953	12.7974	12.2388	0.2741	0.281	0.2762
distilroberta-base euclidean z base	0.4571	0.4292	0.4477	5.8245	6.1945	5.9193	0.4246	0.4307	0.428
distilroberta-base euclidean z f-norm	0.4263	0.4057	0.4176	2.2488	2.4064	2.3201	0.3707	0.3783	0.3737
distilroberta-base euclidean z f-uniform	0.4276	0.4028	0.4179	2.1641	2.3431	2.2485	0.3669	0.3751	0.3718
distilroberta-base euclidean z MP	0.4361	0.4132	0.4285	12.1242	13.9017	12.6855	0.2732	0.2778	0.2756
distilroberta-base euclidean z f-norm + MP	0.4007	0.3794	0.3912	1.5572	1.7217	1.6289	0.3162	0.3212	0.3182
distilroberta-base euclidean z local scaling	0.4541	0.4253	0.445	11.473	12.6273	11.8082	0.2818	0.2859	0.2841

Figure S.2. Error rate and hubness on the test split of the 20 Newsgroups dataset for the medium base model. Means are over the 12 random seeds. error: Error rate, skew: K-skewness, rh: Robinhood score.



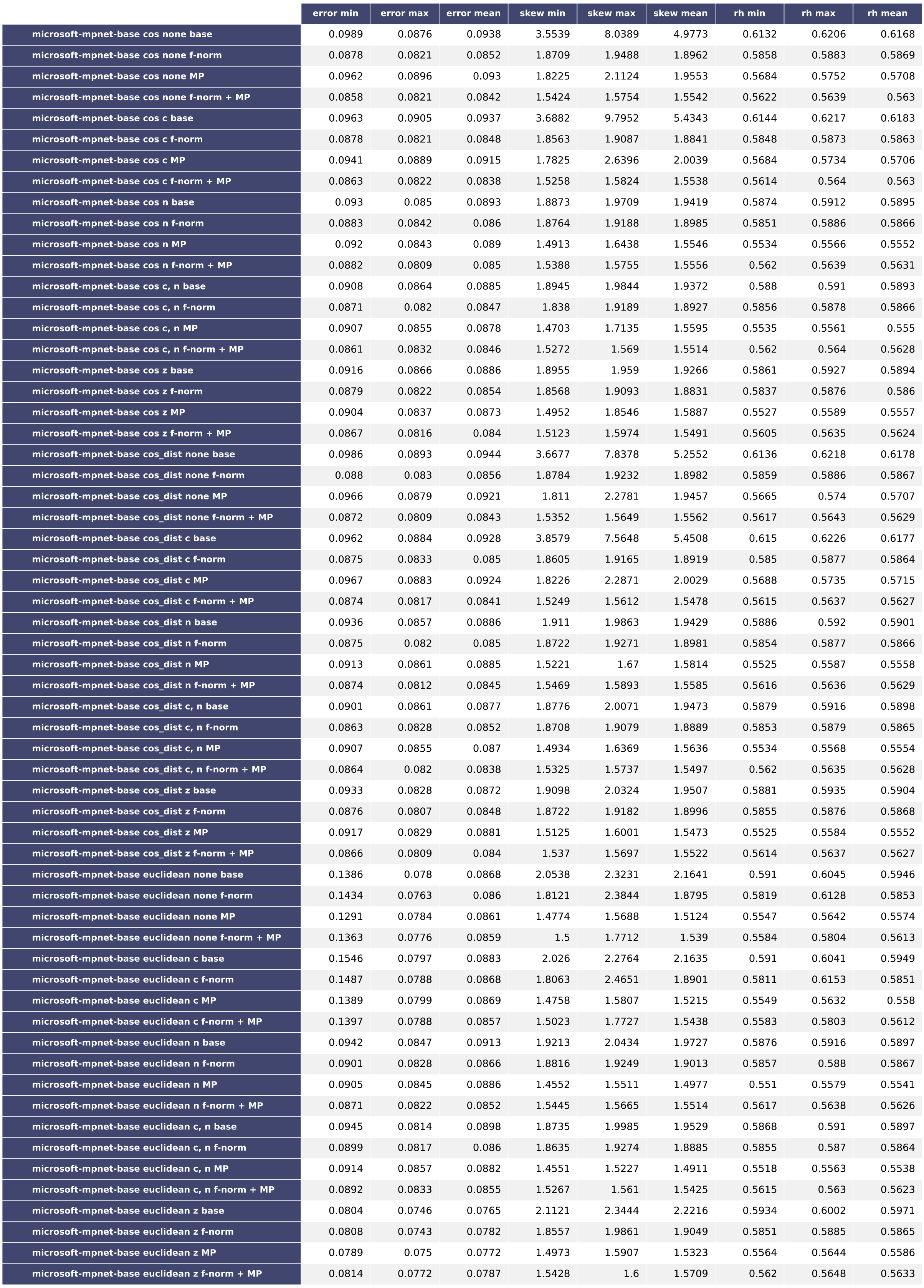
	error min	error max	error mean	skew min	skew max	skew mean	rh min	rh max	rh mean
microsoft-mpnet-base cos none base	0.3991	0.3707	0.3805	4.6421	5.0674	4.7892	0.5742	0.5865	0.5817
microsoft-mpnet-base cos none f-norm	0.3808	0.3633	0.371	2.415	2.694	2.5091	0.3734	0.3829	0.3791
microsoft-mpnet-base cos none f-uniform	0.3747	0.3594	0.3685	2.4072	2.5523	2.4901	0.375	0.385	0.3807
microsoft-mpnet-base cos none MP	0.3581	0.3463	0.3542	7.2927	8.0252	7.5093	0.3767	0.3861	0.3819
microsoft-mpnet-base cos none f-norm + MP	0.3467	0.3364	0.3411	1.8425	2.2684	1.9602	0.3231	0.3295	0.3269
microsoft-mpnet-base cos none local scaling	0.3951	0.3615	0.376	7.0549	7.5638	7.277	0.3827	0.3937	0.3892
microsoft-mpnet-base cos c base	0.3869	0.3702	0.3771	4.6395	5.0612	4.7647	0.5752	0.5861	0.5808
microsoft-mpnet-base cos c f-norm	0.3797	0.3586	0.3707	2.4458	2.6336	2.5369	0.3778	0.3867	0.3813
microsoft-mpnet-base cos c f-uniform	0.3724	0.3611	0.3684	2.4415	2.601	2.5099	0.3786	0.3873	0.3831
microsoft-mpnet-base cos c MP	0.3603	0.3467	0.3532	7.2196	8.3325	7.7173	0.3771	0.386	0.3809
microsoft-mpnet-base cos c f-norm + MP	0.3504	0.3347	0.3415	1.8984	2.0981	1.9552	0.3255	0.3337	0.3291
microsoft-mpnet-base cos c local scaling	0.3816	0.3656	0.373	6.9386	7.925	7.4075	0.3823	0.3939	0.3882
microsoft-mpnet-base cos n base	0.3927	0.376	0.3834	4.9487	5.2228	5.0993	0.466	0.487	0.4756
microsoft-mpnet-base cos n f-norm	0.3842	0.3602	0.3734	2.3144	2.4984	2.4028	0.3668	0.3762	0.3721
microsoft-mpnet-base cos n f-uniform	0.381	0.3671	0.3744	2.3124	2.4695	2.3848	0.3706	0.3783	0.3747
microsoft-mpnet-base cos n MP	0.3579	0.3492	0.3551	10.2147	11.0367	10.5374	0.3014	0.3083	0.305
microsoft-mpnet-base cos n f-norm + MP	0.3516	0.3403	0.3468	1.7875	1.9531	1.8459	0.3183	0.3267	0.3221
microsoft-mpnet-base cos n local scaling	0.3828	0.3668	0.3752	10.7559	11.6309	11.1978	0.2913	0.3004	0.2956
microsoft-mpnet-base cos c, n base	0.3906	0.3654	0.3784	4.9816	5.2412	5.1141	0.463	0.4905	0.4766
microsoft-mpnet-base cos c, n f-norm	0.374	0.3574	0.3677	2.3515	2.6204	2.4494	0.3709	0.3761	0.3734
microsoft-mpnet-base cos c, n f-uniform	0.3763	0.3626	0.369	2.3302	2.6398	2.4675	0.3729	0.3792	0.3753
microsoft-mpnet-base cos c, n MP	0.3553	0.3419	0.3492	10.2407	11.2309	10.635	0.2997	0.3124	0.3057
microsoft-mpnet-base cos c, n f-norm + MP	0.3493	0.3394	0.3446	1.7831	1.9449	1.8707	0.3212	0.3252	0.323
microsoft-mpnet-base cos c, n local scaling	0.3752	0.3615	0.3713	10.8642	11.7308	11.2817	0.2897	0.302	0.2964
microsoft-mpnet-base cos z base	0.4019	0.3707	0.3823	4.9626	5.261	5.1148	0.4642	0.488	0.4736
microsoft-mpnet-base cos z f-norm	0.381	0.3636	0.3713	2.409	2.5241	2.4543	0.3711	0.3785	0.375
microsoft-mpnet-base cos z f-uniform	0.3792	0.3614	0.37	2.3699	2.5372	2.4486	0.3724	0.3801	0.3773
microsoft-mpnet-base cos z MP	0.3586	0.3453	0.3517	10.0851	10.7621	10.4518	0.3017	0.3144	0.3061
microsoft-mpnet-base cos z f-norm + MP	0.3553	0.3338	0.3444	1.8476	1.9589	1.9027	0.3215	0.3262	0.3239
microsoft-mpnet-base cos z local scaling	0.3845	0.3636	0.3715	10.4157	11.7407	11.3447	0.2923	0.3069	0.2967
microsoft-mpnet-base cos_dist none base	0.3889	0.3728	0.3788	4.6297	4.9348	4.7763	0.5797	0.5879	0.5843
microsoft-mpnet-base cos_dist none f-norm	0.3842	0.3617	0.369	2.3662	2.5796	2.4835	0.3723	0.3827	0.3789
microsoft-mpnet-base cos_dist none f-uniform	0.3806	0.3609	0.3689	2.3663	2.5513	2.4623	0.3751	0.3846	0.3812
microsoft-mpnet-base cos_dist none MP	0.3613	0.3505	0.3559	7.2454	7.9397	7.5016	0.3803	0.3866	0.383
microsoft-mpnet-base cos_dist none f-norm + MP	0.3513	0.3362	0.3431	1.8374	2.0664	1.9139	0.3229	0.3308	0.3274
microsoft-mpnet-base cos_dist none local scaling	0.3886	0.3704	0.3753	7.0341	7.339	7.1842	0.3864	0.3927	0.3901
microsoft-mpnet-base cos_dist c base	0.3865	0.371	0.378	4.6	5.0583	4.799	0.5768	0.5972	0.5829
microsoft-mpnet-base cos_dist c f-norm	0.3802	0.3605	0.3695	2.4636	2.7522	2.5578	0.3778	0.3847	0.3804
microsoft-mpnet-base cos_dist c f-uniform	0.3743	0.3606	0.3687	2.441	2.5973	2.5162	0.3777	0.3874	0.3821
microsoft-mpnet-base cos_dist c MP	0.3635	0.3488	0.3555	7.3762	8.2713	7.6956	0.3774	0.3956	0.3831
microsoft-mpnet-base cos_dist c f-norm + MP	0.3485	0.3346	0.3422	1.8871	2.4555	1.9895	0.3267	0.3305	0.3287
microsoft-mpnet-base cos_dist c local scaling	0.3867	0.3643	0.3738	7.0588	7.628	7.3718	0.3829	0.402	0.3898
microsoft-mpnet-base cos_dist n base	0.3894	0.3736	0.3812	4.9994	5.3125	5.1496	0.4628	0.4812	0.4729
microsoft-mpnet-base cos_dist n f-norm	0.3833	0.3621	0.3702	2.3718	2.4875	2.4381	0.3704	0.3774	0.3733
microsoft-mpnet-base cos_dist n f-uniform	0.383	0.3626	0.3718	2.3567	2.4943	2.4307	0.3712	0.3783	0.3754
microsoft-mpnet-base cos_dist n MP	0.3621	0.3448	0.3534	10.2996	11.7678	10.6513	0.3019	0.3088	0.3054
microsoft-mpnet-base cos_dist n f-norm + MP	0.3573	0.3388	0.3472	1.7803	1.9175	1.8663	0.3219	0.3255	0.3231
microsoft-mpnet-base cos_dist n local scaling	0.3829	0.3654	0.3738	10.8398	11.7375	11.2352	0.2929	0.298	0.2956
microsoft-mpnet-base cos_dist c, n base	0.3838	0.3721	0.3787	4.9458	5.159	5.075	0.4685	0.4906	0.479
microsoft-mpnet-base cos_dist c, n f-norm	0.3756	0.3633	0.3702	2.3643	2.5445	2.4415	0.3697	0.3772	0.3738
microsoft-mpnet-base cos_dist c, n f-uniform	0.3776	0.3629	0.37	2.3275	2.5156	2.437	0.3725	0.3805	0.3766
microsoft-mpnet-base cos_dist c, n MP	0.3577	0.3431	0.3503	10.0305	11.8208	10.6775	0.3033	0.3128	0.3072
microsoft-mpnet-base cos_dist c, n f-norm + MP	0.3489	0.3371	0.3423	1.7982	1.9522	1.8806	0.3204	0.326	0.3235
microsoft-mpnet-base cos_dist c, n local scaling	0.3788	0.3583	0.3702	10.4279	12.4252	11.2419	0.2931	0.3049	0.2973
microsoft-mpnet-base cos_dist z base	0.3988	0.3773	0.3843	4.9265	5.1538	5.063	0.4696	0.4907	0.4781
microsoft-mpnet-base cos_dist z f-norm	0.3829	0.3586	0.3709	2.3301	2.5244	2.4286	0.3693	0.3777	0.3736
microsoft-mpnet-base cos_dist z f-uniform	0.3797	0.3626	0.3723	2.3694	2.5189	2.4288	0.3709	0.3799	0.3755
microsoft-mpnet-base cos_dist z MP	0.365	0.3476	0.3547	10.1155	11.5151	10.5876	0.3038	0.3127	0.3074
microsoft-mpnet-base cos_dist z f-norm + MP	0.3533	0.3403	0.3469	1.7746	1.9335	1.8656	0.3183	0.3271	0.3235
microsoft-mpnet-base cos_dist z local scaling	0.3808	0.3651	0.3736	10.5206	12.3096	11.3463	0.2924	0.3046	0.2974
microsoft-mpnet-base euclidean none base	0.7974	0.3644	0.4101	4.7882	7.3648	5.1768	0.3723	0.5412	0.5161
microsoft-mpnet-base euclidean none f-norm	0.7431	0.365	0.4038	2.2137	4.0598	2.4963	0.369	0.392	0.3774
microsoft-mpnet-base euclidean none f-uniform	0.7461	0.3678	0.4053	2.2293	2.6911	2.3681	0.3722	0.3859	0.378
microsoft-mpnet-base euclidean none MP	0.7924	0.3476	0.3917	9.0431	14.7061	9.804	0.2465	0.3392	0.3235
microsoft-mpnet-base euclidean none f-norm + MP	0.733	0.3445	0.3819	1.7488	3.9734	2.0466	0.3138	0.3297	0.3208
microsoft-mpnet-base euclidean none local scaling	0.793	0.3668	0.4092	8.7904	14.2999	9.7001	0.2533	0.3396	0.3261
microsoft-mpnet-base euclidean c base	0.8028	0.3683	0.4139	4.6862	6.6591	5.0654	0.3987	0.5382	0.5165
microsoft-mpnet-base euclidean c f-norm	0.7533	0.365	0.4056	2.2892	5.0678	2.5601	0.3728	0.4216	0.3813
microsoft-mpnet-base euclidean c f-uniform	0.7479	0.3643	0.4044	2.2543	3.8444	2.4383	0.374	0.417	0.382
microsoft-mpnet-base euclidean c MP	0.7937	0.3504	0.3926	9.108	14.2195	9.9638	0.2531	0.3372	0.3229
microsoft-mpnet-base euclidean c f-norm + MP	0.7398	0.3431	0.384	1.783	4.6013	2.0662	0.3178	0.3493	0.3235
microsoft-mpnet-base euclidean c local scaling	0.7953	0.3629	0.4091	8.9108	13.417	9.6902	0.2588	0.3371	0.325
microsoft-mpnet-base euclidean n base	0.3984	0.3698	0.3868	4.9822	5.1932	5.0893	0.4713	0.4839	0.4786
microsoft-mpnet-base euclidean n f-norm	0.3816	0.3607	0.3735	2.3528	2.5421	2.4269	0.3653	0.3782	0.3698
microsoft-mpnet-base euclidean n f-uniform	0.387	0.3643	0.3741	2.3222	2.5688	2.4248	0.3687	0.3784	0.372
microsoft-mpnet-base euclidean n MP	0.369	0.344	0.3565	10.1703	12.1098	10.5925	0.3036	0.3093	0.3068
microsoft-mpnet-base euclidean n f-norm + MP	0.3611	0.338	0.3479	1.7951	1.979	1.8736	0.317	0.3241	0.3197
microsoft-mpnet-base euclidean n local scaling	0.3882	0.3614	0.3783	10.5975	11.3811	10.9881	0.2932	0.3012	0.2973
microsoft-mpnet-base euclidean c, n base	0.3974	0.3867	0.3921	4.9293	5.3214	5.1658	0.464	0.4895	0.4727
microsoft-mpnet-base euclidean c, n f-norm	0.3902	0.3662	0.3788	2.3307	2.5187	2.4228	0.3648	0.377	0.3697
microsoft-mpnet-base euclidean c, n f-uniform	0.3905	0.3642	0.3773	2.3157	2.4975	2.4142	0.3668	0.3768	0.3723
microsoft-mpnet-base euclidean c, n MP	0.3679	0.3465	0.3587	9.967	11.2507	10.6024	0.3002	0.313	0.3043
microsoft-mpnet-base euclidean c, n f-norm + MP	0.3599	0.3417	0.3522	1.7766	1.9882	1.8708	0.3158	0.3251	0.319
microsoft-mpnet-base euclidean c, n local scaling	0.3869	0.3652	0.3805	10.94	12.2089	11.5886	0.2897	0.3059	0.2944
microsoft-mpnet-base euclidean z base	0.4157	0.3829	0.3967	5.2182	5.8085	5.5012	0.4319	0.4697	0.4577
microsoft-mpnet-base euclidean z f-norm	0.4076	0.3826	0.3955	2.2212	4.6847	2.9842	0.3696	0.3821	0.3763
microsoft-mpnet-base euclidean z f-uniform	0.4084	0.384	0.3964	2.11	3.0222	2.3827	0.369	0.3794	0.3741
microsoft-mpnet-base euclidean z MP	0.3978	0.3622	0.3753	11.004	11.9483	11.4027	0.285	0.2984	0.2935
microsoft-mpnet-base euclidean z f-norm + MP	0.3906	0.3574	0.3693	1.6358	5.7345	3.0667	0.3129	0.328	0.3212
microsoft-mpnet-base euclidean z local scaling	0.4125	0.3869	0.3961	10.7203	11.8089	11.2267	0.2911	0.3058	0.2998

Figure S.3. Error rate and hubness on the test split of the 20 Newsgroups dataset for the largest base model tested. Means are over the 12 random seeds. error: Error rate, skew: K-skewness, rh: Robinhood score.



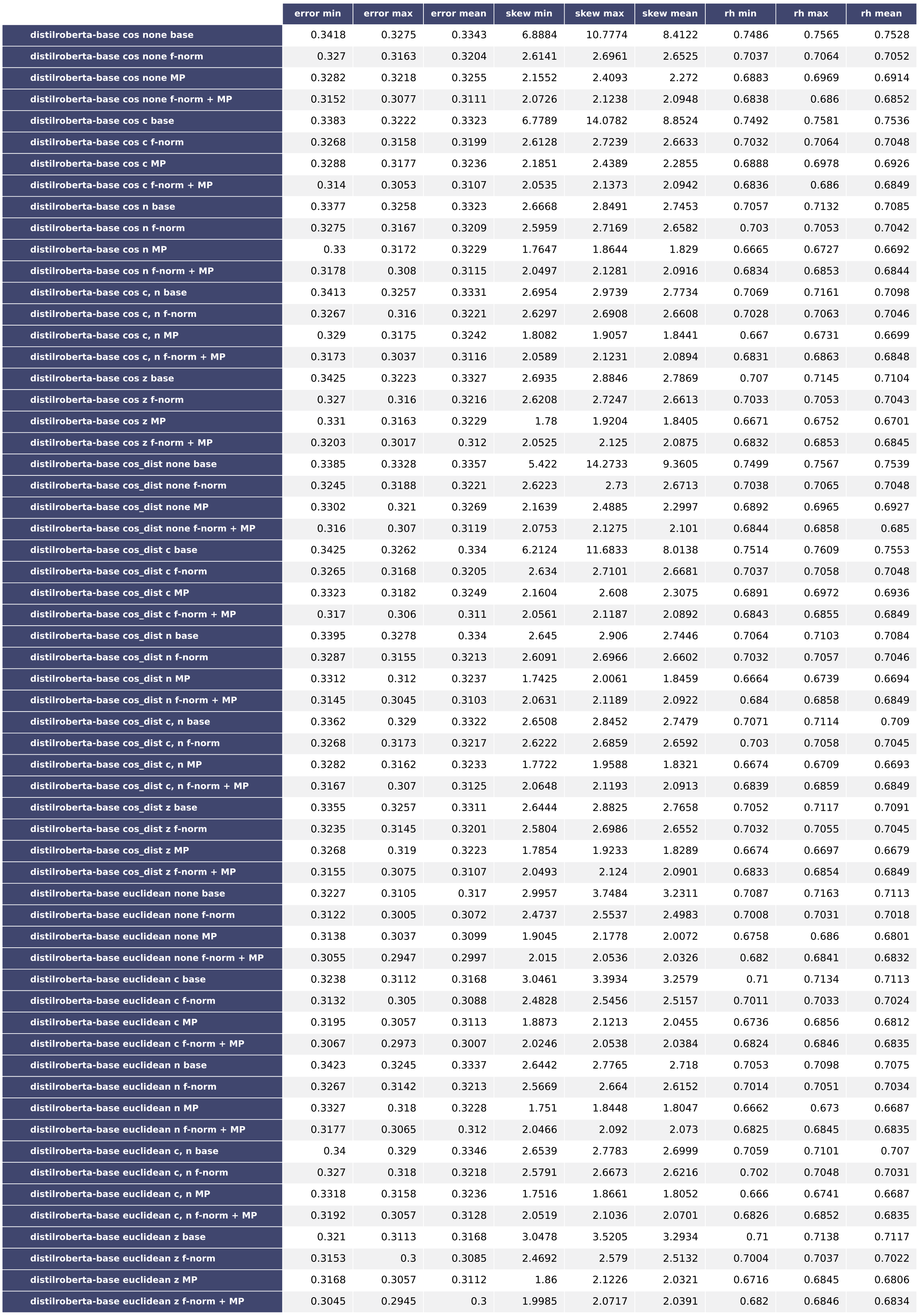
	error min	error max	error mean	skew min	skew max	skew mean	rh min	rh max	rh mean
distilroberta-base cos none base	0.1146	0.1018	0.1085	3.2773	4.0448	3.6443	0.6319	0.6405	0.6365
distilroberta-base cos none f-norm	0.0939	0.0883	0.0906	1.9032	1.9823	1.9492	0.5862	0.5885	0.5876
distilroberta-base cos none MP	0.1103	0.0987	0.1034	1.7707	2.0007	1.8831	0.5726	0.5842	0.5791
distilroberta-base cos none f-norm + MP	0.0909	0.0857	0.088	1.5507	1.5775	1.5603	0.562	0.5642	0.5629
distilroberta-base cos c base	0.1168	0.1046	0.1113	3.3431	4.3386	3.7885	0.6328	0.6458	0.6384
distilroberta-base cos c f-norm	0.0928	0.0878	0.0907	1.9019	2.0092	1.9481	0.5863	0.589	0.5878
distilroberta-base cos c MP	0.1092	0.0999	0.105	1.8325	2.322	1.9558	0.5774	0.5848	0.5806
distilroberta-base cos c f-norm + MP	0.09	0.0855	0.0882	1.5485	1.5987	1.566	0.5615	0.5643	0.563
distilroberta-base cos n base	0.0939	0.0872	0.0898	1.9504	2.0199	1.98	0.5886	0.5921	0.5904
distilroberta-base cos n f-norm	0.0953	0.0868	0.0906	1.904	1.9949	1.946	0.5863	0.5892	0.5876
distilroberta-base cos n MP	0.0911	0.0862	0.0882	1.4639	1.6227	1.527	0.5519	0.5572	0.5541
distilroberta-base cos n f-norm + MP	0.09	0.0833	0.0874	1.544	1.5921	1.564	0.5619	0.5646	0.563
distilroberta-base cos c, n base	0.0938	0.0889	0.0914	1.9509	2.0323	1.9855	0.5896	0.5932	0.5913
distilroberta-base cos c, n f-norm	0.0921	0.087	0.0901	1.9192	1.9853	1.9513	0.5867	0.5887	0.5878
distilroberta-base cos c, n MP	0.0922	0.0871	0.0888	1.4687	1.7822	1.5496	0.5522	0.5562	0.5541
distilroberta-base cos c, n f-norm + MP	0.09	0.0864	0.0881	1.546	1.5776	1.5635	0.5622	0.5645	0.5632
distilroberta-base cos z base	0.0999	0.0887	0.0914	1.9409	2.0187	1.9862	0.5887	0.5941	0.591
distilroberta-base cos z f-norm	0.0947	0.0862	0.0907	1.9218	1.9713	1.9476	0.5868	0.5891	0.5878
distilroberta-base cos z MP	0.093	0.0854	0.0894	1.4665	1.6038	1.5258	0.5511	0.5567	0.5542
distilroberta-base cos z f-norm + MP	0.092	0.085	0.0879	1.5514	1.5835	1.5653	0.562	0.5649	0.5632
distilroberta-base cos_dist none base	0.1187	0.1043	0.1104	3.3292	4.0504	3.6127	0.6359	0.6424	0.6386
distilroberta-base cos_dist none f-norm	0.0939	0.0855	0.0899	1.9235	1.9958	1.9546	0.5859	0.5889	0.5879
distilroberta-base cos_dist none MP	0.1105	0.0974	0.1047	1.8035	2.3054	1.9763	0.5795	0.585	0.5818
distilroberta-base cos_dist none f-norm + MP	0.09	0.0861	0.0879	1.5362	1.6013	1.5633	0.5619	0.5643	0.563
distilroberta-base cos_dist c base	0.1163	0.105	0.1096	3.2641	4.4303	3.6281	0.6319	0.6453	0.6388
distilroberta-base cos_dist c f-norm	0.093	0.0889	0.0908	1.9221	1.9898	1.9476	0.587	0.5889	0.5878
distilroberta-base cos_dist c MP	0.1104	0.0971	0.1043	1.7958	2.3922	1.9761	0.5768	0.5875	0.5816
distilroberta-base cos_dist c f-norm + MP	0.09	0.0857	0.0878	1.55	1.5741	1.5623	0.5621	0.5642	0.5631
distilroberta-base cos_dist n base	0.0925	0.0889	0.0911	1.9261	2.0534	1.9821	0.5888	0.5934	0.591
distilroberta-base cos_dist n f-norm	0.0932	0.0884	0.0906	1.9189	1.9813	1.9466	0.5867	0.5892	0.588
distilroberta-base cos_dist n MP	0.0925	0.0863	0.0894	1.4929	1.5989	1.5283	0.5524	0.5573	0.5543
distilroberta-base cos_dist n f-norm + MP	0.0888	0.0863	0.0875	1.5497	1.5806	1.5632	0.5625	0.5641	0.5634
distilroberta-base cos_dist c, n base	0.0941	0.0866	0.0913	1.9654	2.0164	1.9861	0.5892	0.5921	0.5907
distilroberta-base cos_dist c, n f-norm	0.0925	0.0875	0.0902	1.9149	1.9834	1.9413	0.5857	0.5892	0.5875
distilroberta-base cos_dist c, n MP	0.0888	0.0858	0.0875	1.4609	1.8007	1.5308	0.5511	0.5565	0.5536
distilroberta-base cos_dist c, n f-norm + MP	0.0909	0.0853	0.0874	1.5459	1.5782	1.5642	0.5616	0.5641	0.563
distilroberta-base cos_dist z base	0.0926	0.0875	0.09	1.9394	2.0141	1.9781	0.5896	0.592	0.5906
distilroberta-base cos_dist z f-norm	0.093	0.0891	0.0906	1.9095	1.9817	1.9408	0.5863	0.5883	0.5873
distilroberta-base cos_dist z MP	0.0899	0.0867	0.0885	1.4836	1.6771	1.5389	0.5528	0.5573	0.5549
distilroberta-base cos_dist z f-norm + MP	0.0916	0.0853	0.0881	1.5549	1.5723	1.563	0.5624	0.5635	0.5631
distilroberta-base euclidean none base	0.0821	0.0742	0.0782	2.2288	2.5132	2.3736	0.5938	0.5997	0.5969
distilroberta-base euclidean none f-norm	0.0829	0.0779	0.0802	1.9095	1.9494	1.9312	0.5886	0.5905	0.5893
distilroberta-base euclidean none MP	0.0807	0.0759	0.0784	1.5067	1.628	1.5835	0.5571	0.5624	0.5601
distilroberta-base euclidean none f-norm + MP	0.0846	0.0787	0.0818	1.5676	1.61	1.5868	0.5632	0.5647	0.564
distilroberta-base euclidean c base	0.0824	0.0766	0.0791	2.2898	2.5234	2.4171	0.5952	0.5991	0.5977
distilroberta-base euclidean c f-norm	0.0841	0.0801	0.0818	1.9195	1.9565	1.9418	0.5886	0.5906	0.5895
distilroberta-base euclidean c MP	0.0816	0.0755	0.0786	1.5549	1.6253	1.5849	0.5559	0.5622	0.56
distilroberta-base euclidean c f-norm + MP	0.0855	0.0811	0.0826	1.5769	1.6041	1.5932	0.5633	0.5657	0.5644
distilroberta-base euclidean n base	0.0962	0.0911	0.0938	1.9748	2.0655	2.0181	0.5891	0.5938	0.591
distilroberta-base euclidean n f-norm	0.0951	0.0903	0.0922	1.9218	1.9786	1.9459	0.5862	0.5884	0.5875
distilroberta-base euclidean n MP	0.0922	0.0879	0.0899	1.4748	1.5775	1.5078	0.5526	0.5576	0.5544
distilroberta-base euclidean n f-norm + MP	0.0908	0.0855	0.0882	1.5519	1.5847	1.5684	0.5619	0.5643	0.5629
distilroberta-base euclidean c, n base	0.0968	0.0907	0.0932	1.9468	2.0611	2.0047	0.5883	0.5929	0.5901
distilroberta-base euclidean c, n f-norm	0.0962	0.0896	0.0922	1.8685	1.9661	1.9348	0.5857	0.5882	0.5872
distilroberta-base euclidean c, n MP	0.0939	0.0875	0.0899	1.4486	1.5438	1.4866	0.5515	0.5556	0.5533
distilroberta-base euclidean c, n f-norm + MP	0.0914	0.0854	0.0882	1.5374	1.5845	1.5598	0.5621	0.5638	0.563
distilroberta-base euclidean z base	0.0845	0.0761	0.0796	2.2729	2.5806	2.3958	0.5954	0.6003	0.598
distilroberta-base euclidean z f-norm	0.0845	0.0788	0.0813	1.9063	1.9555	1.9306	0.5875	0.5902	0.5894
distilroberta-base euclidean z MP	0.0817	0.0772	0.0794	1.541	1.5953	1.5667	0.5571	0.563	0.56
distilroberta-base euclidean z f-norm + MP	0.085	0.0797	0.0824	1.5702	1.6007	1.5854	0.5632	0.5654	0.5641

Figure S.4. Error rate and hubness on the test split of the AG News dataset for the medium base model tested. Means are over the 12 random seeds. error: Error rate, skew: K-skewness, rh: Robinhood score.



	error min	error max	error mean	skew min	skew max	skew mean	rh min	rh max	rh mean
microsoft-mpnet-base cos none base	0.0989	0.0876	0.0938	3.5539	8.0389	4.9773	0.6132	0.6206	0.6168
microsoft-mpnet-base cos none f-norm	0.0878	0.0821	0.0852	1.8709	1.9488	1.8962	0.5858	0.5883	0.5869
microsoft-mpnet-base cos none MP	0.0962	0.0896	0.093	1.8225	2.1124	1.9553	0.5684	0.5752	0.5708
microsoft-mpnet-base cos none f-norm + MP	0.0858	0.0821	0.0842	1.5424	1.5754	1.5542	0.5622	0.5639	0.563
microsoft-mpnet-base cos c base	0.0963	0.0905	0.0937	3.6882	9.7952	5.4343	0.6144	0.6217	0.6183
microsoft-mpnet-base cos c f-norm	0.0878	0.0821	0.0848	1.8563	1.9087	1.8841	0.5848	0.5873	0.5863
microsoft-mpnet-base cos c MP	0.0941	0.0889	0.0915	1.7825	2.6396	2.0039	0.5684	0.5734	0.5706
microsoft-mpnet-base cos c f-norm + MP	0.0863	0.0822	0.0838	1.5258	1.5824	1.5538	0.5614	0.564	0.563
microsoft-mpnet-base cos n base	0.093	0.085	0.0893	1.8873	1.9709	1.9419	0.5874	0.5912	0.5895
microsoft-mpnet-base cos n f-norm	0.0883	0.0842	0.086	1.8764	1.9188	1.8985	0.5851	0.5886	0.5866
microsoft-mpnet-base cos n MP	0.092	0.0843	0.089	1.4913	1.6438	1.5546	0.5534	0.5566	0.5552
microsoft-mpnet-base cos n f-norm + MP	0.0882	0.0809	0.085	1.5388	1.5755	1.5556	0.562	0.5639	0.5631
microsoft-mpnet-base cos c, n base	0.0908	0.0864	0.0885	1.8945	1.9844	1.9372	0.588	0.591	0.5893
microsoft-mpnet-base cos c, n f-norm	0.0871	0.082	0.0847	1.838	1.9189	1.8927	0.5856	0.5878	0.5866
microsoft-mpnet-base cos c, n MP	0.0907	0.0855	0.0878	1.4703	1.7135	1.5595	0.5535	0.5561	0.555
microsoft-mpnet-base cos c, n f-norm + MP	0.0861	0.0832	0.0846	1.5272	1.569	1.5514	0.562	0.564	0.5628
microsoft-mpnet-base cos z base	0.0916	0.0866	0.0886	1.8955	1.959	1.9266	0.5861	0.5927	0.5894
microsoft-mpnet-base cos z f-norm	0.0879	0.0822	0.0854	1.8568	1.9093	1.8831	0.5837	0.5876	0.586
microsoft-mpnet-base cos z MP	0.0904	0.0837	0.0873	1.4952	1.8546	1.5887	0.5527	0.5589	0.5557
microsoft-mpnet-base cos z f-norm + MP	0.0867	0.0816	0.084	1.5123	1.5974	1.5491	0.5605	0.5635	0.5624
microsoft-mpnet-base cos_dist none base	0.0986	0.0893	0.0944	3.6677	7.8378	5.2552	0.6136	0.6218	0.6178
microsoft-mpnet-base cos_dist none f-norm	0.088	0.083	0.0856	1.8784	1.9232	1.8982	0.5859	0.5886	0.5867
microsoft-mpnet-base cos_dist none MP	0.0966	0.0879	0.0921	1.811	2.2781	1.9457	0.5665	0.574	0.5707
microsoft-mpnet-base cos_dist none f-norm + MP	0.0872	0.0809	0.0843	1.5352	1.5649	1.5562	0.5617	0.5643	0.5629
microsoft-mpnet-base cos_dist c base	0.0962	0.0884	0.0928	3.8579	7.5648	5.4508	0.615	0.6226	0.6177
microsoft-mpnet-base cos_dist c f-norm	0.0875	0.0833	0.085	1.8605	1.9165	1.8919	0.585	0.5877	0.5864
microsoft-mpnet-base cos_dist c MP	0.0967	0.0883	0.0924	1.8226	2.2871	2.0029	0.5688	0.5735	0.5715
microsoft-mpnet-base cos_dist c f-norm + MP	0.0874	0.0817	0.0841	1.5249	1.5612	1.5478	0.5615	0.5637	0.5627
microsoft-mpnet-base cos_dist n base	0.0936	0.0857	0.0886	1.911	1.9863	1.9429	0.5886	0.592	0.5901
microsoft-mpnet-base cos_dist n f-norm	0.0875	0.082	0.085	1.8722	1.9271	1.8981	0.5854	0.5877	0.5866
microsoft-mpnet-base cos_dist n MP	0.0913	0.0861	0.0885	1.5221	1.67	1.5814	0.5525	0.5587	0.5558
microsoft-mpnet-base cos_dist n f-norm + MP	0.0874	0.0812	0.0845	1.5469	1.5893	1.5585	0.5616	0.5636	0.5629
microsoft-mpnet-base cos_dist c, n base	0.0901	0.0861	0.0877	1.8776	2.0071	1.9473	0.5879	0.5916	0.5898
microsoft-mpnet-base cos_dist c, n f-norm	0.0863	0.0828	0.0852	1.8708	1.9079	1.8889	0.5853	0.5879	0.5865
microsoft-mpnet-base cos_dist c, n MP	0.0907	0.0855	0.087	1.4934	1.6369	1.5636	0.5534	0.5568	0.5554
microsoft-mpnet-base cos_dist c, n f-norm + MP	0.0864	0.082	0.0838	1.5325	1.5737	1.5497	0.562	0.5635	0.5628
microsoft-mpnet-base cos_dist z base	0.0933	0.0828	0.0872	1.9098	2.0324	1.9507	0.5881	0.5935	0.5904
microsoft-mpnet-base cos_dist z f-norm	0.0876	0.0807	0.0848	1.8722	1.9182	1.8996	0.5855	0.5876	0.5868
microsoft-mpnet-base cos_dist z MP	0.0917	0.0829	0.0881	1.5125	1.6001	1.5473	0.5525	0.5584	0.5552
microsoft-mpnet-base cos_dist z f-norm + MP	0.0866	0.0809	0.084	1.537	1.5697	1.5522	0.5614	0.5637	0.5627
microsoft-mpnet-base euclidean none base	0.1386	0.078	0.0868	2.0538	2.3231	2.1641	0.591	0.6045	0.5946
microsoft-mpnet-base euclidean none f-norm	0.1434	0.0763	0.086	1.8121	2.3844	1.8795	0.5819	0.6128	0.5853
microsoft-mpnet-base euclidean none MP	0.1291	0.0784	0.0861	1.4774	1.5688	1.5124	0.5547	0.5642	0.5574
microsoft-mpnet-base euclidean none f-norm + MP	0.1363	0.0776	0.0859	1.5	1.7712	1.539	0.5584	0.5804	0.5613
microsoft-mpnet-base euclidean c base	0.1546	0.0797	0.0883	2.026	2.2764	2.1635	0.591	0.6041	0.5949
microsoft-mpnet-base euclidean c f-norm	0.1487	0.0788	0.0868	1.8063	2.4651	1.8901	0.5811	0.6153	0.5851
microsoft-mpnet-base euclidean c MP	0.1389	0.0799	0.0869	1.4758	1.5807	1.5215	0.5549	0.5632	0.558
microsoft-mpnet-base euclidean c f-norm + MP	0.1397	0.0788	0.0857	1.5023	1.7727	1.5438	0.5583	0.5803	0.5612
microsoft-mpnet-base euclidean n base	0.0942	0.0847	0.0913	1.9213	2.0434	1.9727	0.5876	0.5916	0.5897
microsoft-mpnet-base euclidean n f-norm	0.0901	0.0828	0.0866	1.8816	1.9249	1.9013	0.5857	0.588	0.5867
microsoft-mpnet-base euclidean n MP	0.0905	0.0845	0.0886	1.4552	1.5511	1.4977	0.551	0.5579	0.5541
microsoft-mpnet-base euclidean n f-norm + MP	0.0871	0.0822	0.0852	1.5445	1.5665	1.5514	0.5617	0.5638	0.5626
microsoft-mpnet-base euclidean c, n base	0.0945	0.0814	0.0898	1.8735	1.9985	1.9529	0.5868	0.591	0.5897
microsoft-mpnet-base euclidean c, n f-norm	0.0899	0.0817	0.086	1.8635	1.9274	1.8885	0.5855	0.587	0.5864
microsoft-mpnet-base euclidean c, n MP	0.0914	0.0857	0.0882	1.4551	1.5227	1.4911	0.5518	0.5563	0.5538
microsoft-mpnet-base euclidean c, n f-norm + MP	0.0892	0.0833	0.0855	1.5267	1.561	1.5425	0.5615	0.563	0.5623
microsoft-mpnet-base euclidean z base	0.0804	0.0746	0.0765	2.1121	2.3444	2.2216	0.5934	0.6002	0.5971
microsoft-mpnet-base euclidean z f-norm	0.0808	0.0743	0.0782	1.8557	1.9861	1.9049	0.5851	0.5885	0.5865
microsoft-mpnet-base euclidean z MP	0.0789	0.075	0.0772	1.4973	1.5907	1.5323	0.5564	0.5644	0.5586
microsoft-mpnet-base euclidean z f-norm + MP	0.0814	0.0772	0.0787	1.5428	1.6	1.5709	0.562	0.5648	0.5633

Figure S.5. Error rate and hubness on the test split of the AG News dataset for the largest base model tested. Means are over the 12 random seeds. error: Error rate, skew: K-skewness, rh: Robinhood score.



	error min	error max	error mean	skew min	skew max	skew mean	rh min	rh max	rh mean
distilroberta-base cos none base	0.3418	0.3275	0.3343	6.8884	10.7774	8.4122	0.7486	0.7565	0.7528
distilroberta-base cos none f-norm	0.327	0.3163	0.3204	2.6141	2.6961	2.6525	0.7037	0.7064	0.7052
distilroberta-base cos none MP	0.3282	0.3218	0.3255	2.1552	2.4093	2.272	0.6883	0.6969	0.6914
distilroberta-base cos none f-norm + MP	0.3152	0.3077	0.3111	2.0726	2.1238	2.0948	0.6838	0.686	0.6852
distilroberta-base cos c base	0.3383	0.3222	0.3323	6.7789	14.0782	8.8524	0.7492	0.7581	0.7536
distilroberta-base cos c f-norm	0.3268	0.3158	0.3199	2.6128	2.7239	2.6633	0.7032	0.7064	0.7048
distilroberta-base cos c MP	0.3288	0.3177	0.3236	2.1851	2.4389	2.2855	0.6888	0.6978	0.6926
distilroberta-base cos c f-norm + MP	0.314	0.3053	0.3107	2.0535	2.1373	2.0942	0.6836	0.686	0.6849
distilroberta-base cos n base	0.3377	0.3258	0.3323	2.6668	2.8491	2.7453	0.7057	0.7132	0.7085
distilroberta-base cos n f-norm	0.3275	0.3167	0.3209	2.5959	2.7169	2.6582	0.703	0.7053	0.7042
distilroberta-base cos n MP	0.33	0.3172	0.3229	1.7647	1.8644	1.829	0.6665	0.6727	0.6692
distilroberta-base cos n f-norm + MP	0.3178	0.308	0.3115	2.0497	2.1281	2.0916	0.6834	0.6853	0.6844
distilroberta-base cos c, n base	0.3413	0.3257	0.3331	2.6954	2.9739	2.7734	0.7069	0.7161	0.7098
distilroberta-base cos c, n f-norm	0.3267	0.316	0.3221	2.6297	2.6908	2.6608	0.7028	0.7063	0.7046
distilroberta-base cos c, n MP	0.329	0.3175	0.3242	1.8082	1.9057	1.8441	0.667	0.6731	0.6699
distilroberta-base cos c, n f-norm + MP	0.3173	0.3037	0.3116	2.0589	2.1231	2.0894	0.6831	0.6863	0.6848
distilroberta-base cos z base	0.3425	0.3223	0.3327	2.6935	2.8846	2.7869	0.707	0.7145	0.7104
distilroberta-base cos z f-norm	0.327	0.316	0.3216	2.6208	2.7247	2.6613	0.7033	0.7053	0.7043
distilroberta-base cos z MP	0.331	0.3163	0.3229	1.78	1.9204	1.8405	0.6671	0.6752	0.6701
distilroberta-base cos z f-norm + MP	0.3203	0.3017	0.312	2.0525	2.125	2.0875	0.6832	0.6853	0.6845
distilroberta-base cos_dist none base	0.3385	0.3328	0.3357	5.422	14.2733	9.3605	0.7499	0.7567	0.7539
distilroberta-base cos_dist none f-norm	0.3245	0.3188	0.3221	2.6223	2.73	2.6713	0.7038	0.7065	0.7048
distilroberta-base cos_dist none MP	0.3302	0.321	0.3269	2.1639	2.4885	2.2997	0.6892	0.6965	0.6927
distilroberta-base cos_dist none f-norm + MP	0.316	0.307	0.3119	2.0753	2.1275	2.101	0.6844	0.6858	0.685
distilroberta-base cos_dist c base	0.3425	0.3262	0.334	6.2124	11.6833	8.0138	0.7514	0.7609	0.7553
distilroberta-base cos_dist c f-norm	0.3265	0.3168	0.3205	2.634	2.7101	2.6681	0.7037	0.7058	0.7048
distilroberta-base cos_dist c MP	0.3323	0.3182	0.3249	2.1604	2.608	2.3075	0.6891	0.6972	0.6936
distilroberta-base cos_dist c f-norm + MP	0.317	0.306	0.311	2.0561	2.1187	2.0892	0.6843	0.6855	0.6849
distilroberta-base cos_dist n base	0.3395	0.3278	0.334	2.645	2.906	2.7446	0.7064	0.7103	0.7084
distilroberta-base cos_dist n f-norm	0.3287	0.3155	0.3213	2.6091	2.6966	2.6602	0.7032	0.7057	0.7046
distilroberta-base cos_dist n MP	0.3312	0.312	0.3237	1.7425	2.0061	1.8459	0.6664	0.6739	0.6694
distilroberta-base cos_dist n f-norm + MP	0.3145	0.3045	0.3103	2.0631	2.1189	2.0922	0.684	0.6858	0.6849
distilroberta-base cos_dist c, n base	0.3362	0.329	0.3322	2.6508	2.8452	2.7479	0.7071	0.7114	0.709
distilroberta-base cos_dist c, n f-norm	0.3268	0.3173	0.3217	2.6222	2.6859	2.6592	0.703	0.7058	0.7045
distilroberta-base cos_dist c, n MP	0.3282	0.3162	0.3233	1.7722	1.9588	1.8321	0.6674	0.6709	0.6693
distilroberta-base cos_dist c, n f-norm + MP	0.3167	0.307	0.3125	2.0648	2.1193	2.0913	0.6839	0.6859	0.6849
distilroberta-base cos_dist z base	0.3355	0.3257	0.3311	2.6444	2.8825	2.7658	0.7052	0.7117	0.7091
distilroberta-base cos_dist z f-norm	0.3235	0.3145	0.3201	2.5804	2.6986	2.6552	0.7032	0.7055	0.7045
distilroberta-base cos_dist z MP	0.3268	0.319	0.3223	1.7854	1.9233	1.8289	0.6674	0.6697	0.6679
distilroberta-base cos_dist z f-norm + MP	0.3155	0.3075	0.3107	2.0493	2.124	2.0901	0.6833	0.6854	0.6849
distilroberta-base euclidean none base	0.3227	0.3105	0.317	2.9957	3.7484	3.2311	0.7087	0.7163	0.7113
distilroberta-base euclidean none f-norm	0.3122	0.3005	0.3072	2.4737	2.5537	2.4983	0.7008	0.7031	0.7018
distilroberta-base euclidean none MP	0.3138	0.3037	0.3099	1.9045	2.1778	2.0072	0.6758	0.686	0.6801
distilroberta-base euclidean none f-norm + MP	0.3055	0.2947	0.2997	2.015	2.0536	2.0326	0.682	0.6841	0.6832
distilroberta-base euclidean c base	0.3238	0.3112	0.3168	3.0461	3.3934	3.2579	0.71	0.7134	0.7113
distilroberta-base euclidean c f-norm	0.3132	0.305	0.3088	2.4828	2.5456	2.5157	0.7011	0.7033	0.7024
distilroberta-base euclidean c MP	0.3195	0.3057	0.3113	1.8873	2.1213	2.0455	0.6736	0.6856	0.6812
distilroberta-base euclidean c f-norm + MP	0.3067	0.2973	0.3007	2.0246	2.0538	2.0384	0.6824	0.6846	0.6835
distilroberta-base euclidean n base	0.3423	0.3245	0.3337	2.6442	2.7765	2.718	0.7053	0.7098	0.7075
distilroberta-base euclidean n f-norm	0.3267	0.3142	0.3213	2.5669	2.664	2.6152	0.7014	0.7051	0.7034
distilroberta-base euclidean n MP	0.3327	0.318	0.3228	1.751	1.8448	1.8047	0.6662	0.673	0.6687
distilroberta-base euclidean n f-norm + MP	0.3177	0.3065	0.312	2.0466	2.092	2.073	0.6825	0.6845	0.6835
distilroberta-base euclidean c, n base	0.34	0.329	0.3346	2.6539	2.7783	2.6999	0.7059	0.7101	0.707
distilroberta-base euclidean c, n f-norm	0.327	0.318	0.3218	2.5791	2.6673	2.6216	0.702	0.7048	0.7031
distilroberta-base euclidean c, n MP	0.3318	0.3158	0.3236	1.7516	1.8661	1.8052	0.666	0.6741	0.6687
distilroberta-base euclidean c, n f-norm + MP	0.3192	0.3057	0.3128	2.0519	2.1036	2.0701	0.6826	0.6852	0.6835
distilroberta-base euclidean z base	0.321	0.3113	0.3168	3.0478	3.5205	3.2934	0.71	0.7138	0.7117
distilroberta-base euclidean z f-norm	0.3153	0.3	0.3085	2.4692	2.579	2.5132	0.7004	0.7037	0.7022
distilroberta-base euclidean z MP	0.3168	0.3057	0.3112	1.86	2.1226	2.0321	0.6716	0.6845	0.6806
distilroberta-base euclidean z f-norm + MP	0.3045	0.2945	0.3	1.9985	2.0717	2.0391	0.682	0.6846	0.6834

Figure S.6. Error rate and hubness on the test split of the ten percent of the Yahoo Answers dataset for the medium base model tested. Means are over the 12 random seeds. error: Error rate, skew: K-skewness, rh: Robinhood score.

	error min	error max	error mean	skew min	skew max	skew mean	rh min	rh max	rh mean
microsoft-mpnet-base cos none base	0.3345	0.3167	0.3246	4.9833	6.3753	5.5384	0.7792	0.7888	0.7846
microsoft-mpnet-base cos none f-norm	0.3143	0.3023	0.3068	2.6058	2.7308	2.6669	0.7038	0.706	0.7048
microsoft-mpnet-base cos none MP	0.3207	0.305	0.3123	2.3466	2.6131	2.4763	0.7046	0.7135	0.7091
microsoft-mpnet-base cos none f-norm + MP	0.3043	0.2948	0.2986	2.1064	2.157	2.1228	0.6847	0.6874	0.686
microsoft-mpnet-base cos c base	0.328	0.3158	0.3208	4.8838	6.0819	5.5615	0.779	0.7882	0.7839
microsoft-mpnet-base cos c f-norm	0.31	0.2992	0.3043	2.6318	2.7231	2.6684	0.7043	0.7057	0.705
microsoft-mpnet-base cos c MP	0.32	0.3068	0.3114	2.311	2.7169	2.4988	0.7031	0.7177	0.7093
microsoft-mpnet-base cos c f-norm + MP	0.3045	0.2925	0.2971	2.1071	2.1414	2.1237	0.6852	0.6869	0.6862
microsoft-mpnet-base cos n base	0.3283	0.319	0.3236	3.1271	3.4941	3.3399	0.7225	0.7306	0.7268
microsoft-mpnet-base cos n f-norm	0.3138	0.303	0.3081	2.6174	2.6932	2.6599	0.7035	0.705	0.7043
microsoft-mpnet-base cos n MP	0.32	0.3028	0.3119	1.8178	1.9608	1.899	0.6717	0.6779	0.6745
microsoft-mpnet-base cos n f-norm + MP	0.3015	0.2953	0.2989	2.0937	2.1419	2.1208	0.6851	0.6869	0.6859
microsoft-mpnet-base cos c, n base	0.3252	0.3157	0.3206	3.1354	3.529	3.3142	0.7211	0.7309	0.7264
microsoft-mpnet-base cos c, n f-norm	0.3107	0.3013	0.3065	2.6327	2.7147	2.672	0.7033	0.7052	0.7045
microsoft-mpnet-base cos c, n MP	0.3165	0.3042	0.3107	1.8496	1.9886	1.8891	0.6722	0.681	0.6747
microsoft-mpnet-base cos c, n f-norm + MP	0.3028	0.2948	0.2993	2.0919	2.1398	2.1212	0.6848	0.6867	0.6858
microsoft-mpnet-base cos z base	0.327	0.315	0.3211	3.0703	3.5847	3.2688	0.7212	0.7327	0.7258
microsoft-mpnet-base cos z f-norm	0.3128	0.3002	0.3065	2.6134	2.6993	2.6621	0.7032	0.7053	0.7045
microsoft-mpnet-base cos z MP	0.3157	0.3025	0.3105	1.8156	2.0291	1.9057	0.671	0.6826	0.6752
microsoft-mpnet-base cos z f-norm + MP	0.3057	0.2955	0.2992	2.1015	2.1488	2.1212	0.6848	0.6869	0.6858
microsoft-mpnet-base cos_dist none base	0.331	0.3172	0.3228	5.2076	6.484	5.6656	0.7818	0.7898	0.7858
microsoft-mpnet-base cos_dist none f-norm	0.3093	0.302	0.3059	2.6514	2.7325	2.6806	0.7036	0.7062	0.7048
microsoft-mpnet-base cos_dist none MP	0.3182	0.3065	0.3124	2.3971	2.7527	2.5044	0.7048	0.7211	0.7105
microsoft-mpnet-base cos_dist none f-norm + MP	0.3018	0.2917	0.2974	2.0932	2.1548	2.1281	0.6847	0.6877	0.686
microsoft-mpnet-base cos_dist c base	0.33	0.315	0.322	5.3192	6.2058	5.639	0.7822	0.7942	0.7861
microsoft-mpnet-base cos_dist c f-norm	0.3087	0.2993	0.3055	2.6421	2.7404	2.6906	0.7041	0.7072	0.7051
microsoft-mpnet-base cos_dist c MP	0.318	0.3095	0.3123	2.3167	2.6398	2.5263	0.7059	0.7167	0.7117
microsoft-mpnet-base cos_dist c f-norm + MP	0.3047	0.2937	0.2977	2.1071	2.1545	2.1328	0.6855	0.6878	0.6863
microsoft-mpnet-base cos_dist n base	0.326	0.3143	0.3224	3.2091	3.4447	3.3405	0.7236	0.7286	0.7265
microsoft-mpnet-base cos_dist n f-norm	0.3125	0.3017	0.3074	2.6228	2.7628	2.683	0.7035	0.7058	0.7047
microsoft-mpnet-base cos_dist n MP	0.3218	0.3097	0.3143	1.8312	1.9531	1.8919	0.6715	0.6779	0.6748
microsoft-mpnet-base cos_dist n f-norm + MP	0.3042	0.295	0.3001	2.0945	2.179	2.1319	0.6851	0.6868	0.686
microsoft-mpnet-base cos_dist c, n base	0.3248	0.3157	0.3208	3.1389	3.6588	3.3294	0.7227	0.7351	0.7273
microsoft-mpnet-base cos_dist c, n f-norm	0.3122	0.2973	0.3057	2.5943	2.6835	2.6571	0.703	0.7056	0.7039
microsoft-mpnet-base cos_dist c, n MP	0.3157	0.3012	0.3107	1.8201	1.9797	1.8922	0.6719	0.6771	0.6741
microsoft-mpnet-base cos_dist c, n f-norm + MP	0.3008	0.2893	0.2975	2.0933	2.1406	2.1215	0.6845	0.6864	0.6854
microsoft-mpnet-base cos_dist z base	0.324	0.3138	0.3189	3.1493	3.6177	3.3176	0.7228	0.7334	0.7267
microsoft-mpnet-base cos_dist z f-norm	0.3112	0.2997	0.3056	2.6184	2.7031	2.663	0.7034	0.705	0.7043
microsoft-mpnet-base cos_dist z MP	0.3155	0.3028	0.3102	1.8282	2.0795	1.9068	0.671	0.6882	0.6752
microsoft-mpnet-base cos_dist z f-norm + MP	0.3022	0.2922	0.2982	2.0955	2.1395	2.1223	0.6852	0.6866	0.6857
microsoft-mpnet-base euclidean none base	0.5467	0.2918	0.3184	2.6009	6.0489	5.023	0.7026	0.7485	0.7395
microsoft-mpnet-base euclidean none f-norm	0.521	0.29	0.3134	2.4585	3.1904	2.5585	0.698	0.7073	0.7003
microsoft-mpnet-base euclidean none MP	0.5435	0.2813	0.3101	1.8927	2.0359	1.9716	0.6742	0.6828	0.6795
microsoft-mpnet-base euclidean none f-norm + MP	0.5117	0.2835	0.3053	1.9828	2.3329	2.0428	0.6794	0.6886	0.6813
microsoft-mpnet-base euclidean c base	0.5945	0.2897	0.3213	2.7046	5.5758	4.8188	0.7058	0.7489	0.7397
microsoft-mpnet-base euclidean c f-norm	0.5595	0.2885	0.3169	2.4737	3.4998	2.6005	0.699	0.7173	0.7014
microsoft-mpnet-base euclidean c MP	0.5777	0.2813	0.3138	1.9009	2.0508	1.9619	0.6756	0.6837	0.6795
microsoft-mpnet-base euclidean c f-norm + MP	0.5453	0.2828	0.3075	2.0007	2.5171	2.0615	0.6801	0.697	0.6822
microsoft-mpnet-base euclidean n base	0.3333	0.3167	0.3242	3.155	3.4475	3.3065	0.7217	0.7288	0.7262
microsoft-mpnet-base euclidean n f-norm	0.3163	0.309	0.3127	2.5844	2.6886	2.6275	0.7025	0.7054	0.7039
microsoft-mpnet-base euclidean n MP	0.322	0.3065	0.314	1.8253	1.9794	1.8872	0.6722	0.6793	0.6746
microsoft-mpnet-base euclidean n f-norm + MP	0.3107	0.297	0.3028	2.0845	2.1448	2.1123	0.6844	0.6863	0.6851
microsoft-mpnet-base euclidean c, n base	0.3272	0.3195	0.3233	3.053	3.6123	3.2205	0.7185	0.7312	0.7232
microsoft-mpnet-base euclidean c, n f-norm	0.3155	0.3042	0.3107	2.5758	2.6488	2.6169	0.7029	0.705	0.7038
microsoft-mpnet-base euclidean c, n MP	0.3187	0.3058	0.3129	1.8081	1.9537	1.8436	0.67	0.6781	0.6718
microsoft-mpnet-base euclidean c, n f-norm + MP	0.307	0.2967	0.303	2.0805	2.1307	2.0995	0.6843	0.686	0.685
microsoft-mpnet-base euclidean z base	0.2993	0.2878	0.2942	2.7443	3.3269	3.0544	0.7064	0.7221	0.7152
microsoft-mpnet-base euclidean z f-norm	0.3037	0.2847	0.2904	2.3278	2.5262	2.4654	0.6962	0.7017	0.7001
microsoft-mpnet-base euclidean z MP	0.2945	0.2837	0.2872	1.7611	2.0349	1.8919	0.6665	0.6803	0.6743
microsoft-mpnet-base euclidean z f-norm + MP	0.2942	0.2795	0.2832	1.9545	2.0508	2.0209	0.6777	0.6826	0.6812

Figure S.7. Error rate and hubness on the test split of the ten percent of the Yahoo Answers dataset for the largest base model tested. Means are over the 12 random seeds. error: Error rate, skew: K-skewness, rh: Robinhood score.