Compressing Sentence Representation with Maximum Coding Rate Reduction

Domagoj Ševerdija*, Tomislav Prusina*, Antonio Jovanović*, Luka Borozan*,
Jurica Maltar*, Domagoj Matijević*
* Department of Mathematics, Josip Juraj Strossmayer University of Osijek, Osijek, Croatia
dseverdi@mathos.hr

***Abstract*—In most natural language inference problems, sentence representation is needed for semantic retrieval tasks. In recent years, pre-trained large language models have been quite effective for computing such representations. These models produce high-dimensional sentence embeddings. An evident performance gap between large and small models exists in practice. Hence, due to space and time hardware limitations, there is a need to attain comparable results when using the smaller model, which is usually a distilled version of the large language model. In this paper, we assess the model distillation of the sentence representation model Sentence-BERT by augmenting the pre-trained distilled model with a projection layer additionally learned on the Maximum Coding Rate Reduction (MCR^2) objective, a novel approach developed for general purpose manifold clustering.**

We demonstrate that the new language model with reduced complexity and sentence embedding size can achieve comparable results on semantic retrieval benchmarks.

***Keywords*—Sentence embeddings, model distillation, Maximum Coding Rate Reduction, semantic retrieval**

I. Introduction

Dense vector representations of words, or word embeddings, form the backbone of most NLP applications and can be constructed using context-free (see [1], [2], [3]) or contextualized methods (see [4], [5] for more details). In practice, few NLP applications often benefit from having sentence or document representations in addition to word embeddings. In most cases, one can use the weighted average (aka pooling) over some or all of the word embeddings from a sentence or document. Although it disregards word order while pooling, this approach has shown to be reasonably performant [6]. Pre-trained language models like BERT have shown success on many NLP tasks through fine-tuning. Unfortunately, using contextualized word vectors from these models as a sentence representation is significantly inferior in terms of semantic textual similarity compared to approaches when one uses non-contextualized word vectors, which are trained with a much simpler model (see [7] for more details). Therefore, more sophisticated methods were derived to find efficient and performant universal sentence encoders. Reimers et al. in [7] developed the Sentence-BERT model by fine-tuning pre-trained BERT architecture on sentence pair scoring tasks using a Siamese architecture to learn better sentence representations, showing much improvement in downstream NLP tasks. Their approach ended up with a relatively large model size (hundreds of millions to billions of parameters) and sentence embedding dimension 768, a relatively large number for efficient search and retrieval operations over databases. In this paper, we focus on reducing the dimensionality of sentence embeddings up to 50%-70% while still achieving comparable results across the board of NLP benchmarks. This opens up a possibility of deploying AI models on smaller-scale computer systems like embedded systems.

A. Related Work

Following the distributional hypothesis, Mikolov et al. in [2] showed that computing dense vectors of lower dimension for word embeddings give interesting mathematical properties of words. Inspired by the same idea, Kiros et al. [8] and Lee et al. [9] tried to derive a model which predicts surrounding sentences. Sent2Vec [10] generates context-free sentence embeddings as averages of word vectors and n-gram vectors (similar to FastText [11] for words). Conneau et al. [12] computed contextualized sentence embeddings using a BiLSTM Siamese network that was fine-tuned on pairs of semantically similar sentences. This approach was extended to fine-tuning pre-trained language models like BERT in [7]. Recently, Gao et al. [13] improved this approach by suggesting a contrastive learning method and achieved state-of-the-art results. Projecting sentence embeddings to lower dimensions was motivated by projecting word vectors. In most cases, PCA methods gave surprisingly good results and even retrofitted the word vectors in such a way that it made vectors more isotropic which had a good impact on NLP benchmarks. Li et al. [14] showed that this anomaly is also apparent in sentence vectors and gave a normalizing flow method to retrofit such vectors. Recent work of [15] introduced Maximum Coding Rate Reduction (MCR^2), a novel learning objective that enables for learning a subspace representation given the clustering[1]. They also demonstrated how to extend the approach to the problem of unsupervised clustering.

B. Our contribution

We use a pre-trained sentence embedding model like Sentence-BERT (SBERT) as a sentence encoder and train a non-linear mapper atop the encoder using a Maximal Coding Rate Reduction as a training objective for

[1]The formal definition of MCR^2 will be given in Section II.

learning discriminative low-dimensional structures that preserve all the essential information encoded into the high-dimensional data. This approach allows for more robust training than standard training objectives like cross-entropy and produces clusters in the embedding space. The main contribution of our paper is a sentence embedding compression technique that achieves comparable results with smaller sentence embedding sizes on semantic NLP benchmarks compared to the baseline sentence encoder.

The paper is organized as follows. In Section II we describe Maximum Rate Coding Reduction training objective for computing subspace embedding space. Furthermore, SBERT architecture is described as a sentence encoder followed by a definition of the projection layer. In Section III we experimentally evaluate our method and conclude with a results discussion.

II. Method

For a given set of sentences S and for each sentence

$$(word_1, word_2, \ldots, word_{n_i}) \in S$$

our task is to construct a lower dimensional embedding $z_i \in \mathbb{R}^d$ that contains important semantic information characteristic for that sentence. Our idea is to extend SBERT and from it's embedding compute a small projector to reduce the dimension, i.e. given the set of SBERT's embeddings $Z \in \mathbb{R}^{d\times n}$ of the dataset S, find a $\hat{Z} \in \mathbb{R}^{\hat{d}\times n}$ that retains semantic information extracted by SBERT.

A. Learning a subspace representation with MCR^2

Using the idea from Li et al. [16] we aim to minimize the angle between similar sentences and maximize the entropy of the whole dataset. For two representations $\hat{z}_1, \hat{z}_2 \in \mathbb{R}^d$ of two sentences we measure how similar they are by cosine similarity

$$D\left(\hat{z}_1, \hat{z}_2\right) = \frac{\cos\left(\hat{z}_1^\top \hat{z}_2\right)}{\|\hat{z}_1\|_2 \|\hat{z}_2\|_2}.$$

For two sets $\hat{Z}_1, \hat{Z}_2 \in \mathbb{R}^{d\times b}$ we define this function as

$$D(\hat{Z}_1, \hat{Z}_2) = \frac{1}{b}\sum_{i=1}^{b} D\left(\hat{z}_{1,i}, \hat{z}_{2,i}\right) \tag{1}$$

where $\hat{z}_{1,i}$ is the i-th element of $\hat{Z}_1$ and $\hat{z}_{2,i}$ is the i-th element of $\hat{Z}_2$. Given pairs of similar sentences we want them to have the D score as large as possible.

For a set of representations $\hat{Z} \in \mathbb{R}^{d\times n}$ with n elements, its entropy is defined as

$$R_\varepsilon(\hat{Z}) = \frac{1}{2}\log\det\left(I + \frac{d}{n\varepsilon^2}\hat{Z}\hat{Z}^\top\right) \tag{2}$$

for a given parameter ε and identity matrix I. This function is approximately the Shannon coding rate function for multivariate Gaussian distribution given average distortion ε [17]. Maximizing (2) we maximize the volume of the ball in which the embeddings are packed. The theory behind this is well over the scope of this paper. It is given in the paper by Ma et al. [18] where they explore rate distortion, ε-ball packing and lossy encoding with normally distributed data. By optimizing it in parallel with (1) we try to distance each sentence from others, except for the similar pairs that we try to keep close. Additionally, given cluster assignments, we can measure the entropy of each cluster with

$$R_\varepsilon\left(\hat{Z}, \Pi_k\right) = \frac{n_k}{2n}\log\det\left(I + \frac{d}{n_k\varepsilon^2}\hat{Z}\Pi_k\hat{Z}^\top\right) \tag{3}$$

where Π_k is a diagonal matrix with i-th entry being 1 if the i-th sentence belongs to cluster k, otherwise 0, and $n_k = \mathrm{tr}\left(\Pi_k\right)$, trace of matrix Π_k, i.e. number of points in this cluster. Combining functions (1), (2) and (3) into one we get the MRC^2 loss function defined with

$$L(\hat{Z}, \Pi) = -R_\varepsilon(\hat{Z}) + \sum_{i=1}^{k} R_\varepsilon(\hat{Z}, \Pi_i) - \lambda D(\hat{Z}_1, \hat{Z}_2) \tag{4}$$

for some hyperparameter λ and pairs of similar sentences respectively divided into two sets $\hat{Z}_1, \hat{Z}_2$. Π denoted in $L(\hat{Z}, \Pi)$ is the clustering of data given by the user or learned by the architecture. The choice of λ depends on how close we want to keep similar sentences in our projection. For larger values of λ the network focuses on collapsing similar pairs into the same vector which, if one is not careful enough, can lead to collapsing all vectors into one. For smaller values of λ the network has more freedom to decide which vector embeddings to keep close. This, on the other hand, can lead to an unwanted vector representation that tends to maximally distance vectors from each other. By minimizing (4) we

- maximize the volume of all embeddings, $R_\varepsilon(\hat{Z})$,
- minimize the volume of each cluster, $\sum_{i=1}^{k} R_\varepsilon(\hat{Z}, \Pi_i)$,
- maximize the cosine similarity of pairs of similar sentences, $\lambda D(\hat{Z}_1, \hat{Z}_2)$.

The consequence of this is that after the minimization we have an embedding in which different clusters are orthogonal to each other (see [15] for more details), i.e.

$$i \neq j \implies \hat{Z}_i\hat{Z}_j^\top = 0. \tag{5}$$

B. Architecture

Our model receives as input a batch of sentences S, encodes a sentence representations Z and outputs projected sentence representations $\hat{Z}$ together with cluster assignments Π for S. The overall architecture is shown in Fig. 1.

1) Sentence encoder: BERT [4] and its variants has set a new state-of-the art performance on sentence-pair regression and classification tasks. Unfortunately, it requires that both sentences are fed into network causing a computation overhead which renders simple tasks like finding similar sentence pairs in large datasets a costly procedure. Therefore, SBERT [7] is a modification of the BERT network which uses siamese network that is able to derive semantically meaningful sentence representations.

The model consists of BERT as a pre-trained encoder, a pooling layer that computes sentence representation as an average of hidden states from the last layer of BERT. SBERT is trained on the combination of the SNLI [19] and MultiNLI [20] datasets.

2) Projection layer: Following Li et al. [16] we use above mentioned SBERT as a backbone and two last linear heads used to produce features and cluster logits. Features given by the first head are additionally normalized to unit sphere and the clusters are learned from the given pairs of similar sentences. The whole architecture is described in Fig. 1 where blue denotes the SBERT model and gray denotes a feed forward neural network that we call a projection layer. In this projection layer we have two heads. The first head colored in red is a single linear layer that collects information about the clusters and applies Gumbel-Softmax [21]. The second head colored in green is again a single linear layer that outputs features which are in turn normalized to zero mean and unit variance. The ELU activation function is used due to its good properties [22].

III. Experiments

We trained our model on StackExchange duplicate questions as title/title pairs, used from CQADupStack [23]. The pipeline from Sentence Transformers[2] for SBERT and projection layer was used with default settings, 256 batch size and 50 epochs. The backbone *all-mpnet-base-v2* and distilled model *all-MiniLM-L6-v2* [24] as pre-trained SBERT were frozen and the only trained part was the projection layer. We refer to the former as MPNET and the latter as MiniLM. Hyperparameter λ from equation (4) was set to 2000 for dimensions 50 and 100, and to 4000 for all other dimensions. Our model is evaluated on several downstream NLP tasks. First of all, we test our model on those benchmarks that can include clustering, namely, semantic retrieval tasks. We also show that computed low-dimension sentence representations behave reasonably well on other semantic benchmarks. The sizes of these dimensions are motivated by experimental observation of suitable word vector sizes from [25] and [14] in which a connection between word vectors and sentence embeddings is established. For downstream NLP tasks such as standard textual similarity, sentiment analysis and question-type classification tasks we use available datasets from SentEval evaluation toolkit [26] for sentence embeddings. See [26] and references therein for dataset descriptions.

All our experiments were evaluated on AMD Ryzen Threadripper 3990X 64-Core Processor @ 4.3GHz, Nvidia GeForce RTX 3090 GPU, CUDA 11.6 with PyTorch implementation 1.9.1.

A. Semantic Retrieval (SR) Task

The semantic retrieval (SR) task is to find all sentences in the retrieval corpus that are semantically similar to the query sentence. The basic framework is to compute sentence embeddings for the retrieval corpus and the query sentence. The goal is to find closest points in retrieval corpus embedding space to the query. Sometimes, to speed up the process [27], one can cluster sentences in the retrieval corpus embedding space into k clusters and use query sentence to find the closest cluster of sentences. The Quora Duplicate Question Dataset[3] is used to evaluate our method. This dataset consists of 500k sentences with over 400k annotated question pairs if they are duplicates or not.

B. Semantic Textual Similarity (STS) Task

One of the baseline benchmarks in natural language processing is the semantic textual similarity (STS) task that qualitatively assesses the semantic similarity between two sentences (i.e., text snippets). Our model is evaluated by computing cosine similarity between sentence pair embeddings on standard STS tasks: STS 2012-2016 and STS Benchmark available in SentEval. These datasets were labeled between 0 and 5 scores indicating the semantic relatedness of sentence pairs. Evaluation on these datasets is conducted using Spearman rank correlation which measures the correlation quality between calculated and human labeled similarity. It is valued from -1 and 1 which will be high if the ranks of predicted similarities and human labels are similar.

C. Sentence classification (SC) Task

Sentiment classification tasks involve assigning a score for a sentiment of a snippet of text. It is formulated as a classification of text into two or more sentiment classes, namely negative, positive or neutral, or something in-between. Datasets SST, SUBJ, CR, MR are typical benchmarks for sentiment analysis. Moreover, another example of a sentence classification task is to assign a question type for a question, like in the TREC task. In the paraphrase detection problem (like MRPC), one must classify if one sentence is a paraphrase of the other. MPQA dataset is an example of opinion classification task. The performance metric for these benchmarks is given as accuracy. All these datasets are available in SentEval toolkit.

IV. Results

This section compares our method as a clustering and compression algorithm, respectively. In Table I, we compare how our clustering competes with k-means[4] algorithm and report time performance (needed time for encoding vectors, clustering, and total time) In the second part, we test our compression against semantic relatedness tasks. The results are reported in Tables I, II and III. Model names in these tables are structured as follows: the sBERT pretrained model name, the abrreviation MCR indicates that MCR[2] is used as a projection, the number followed is a projection dimension, and optionally if k-means is used.

[2]https://github.com/UKPLab/sentence-transformers

[3]https://www.kaggle.com/datasets/sambit7/first-quora-dataset

[4]implemented in *scikit-learn* Python package

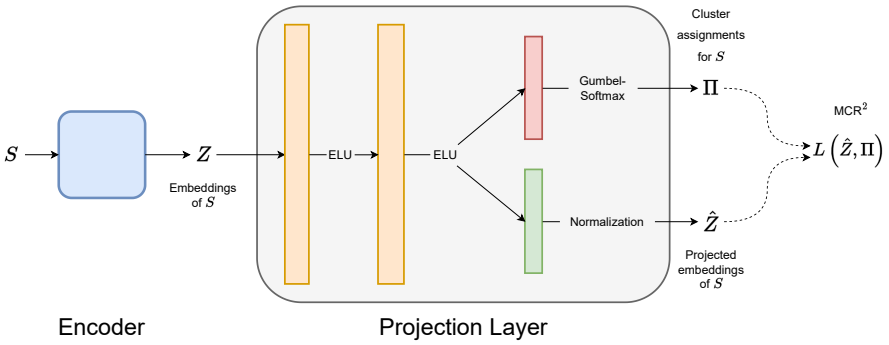

Fig. 1: The overall architecture

TABLE I: For Semantic Retrieval (SR) tasks the `all-mpnet-base-v2` SBERT model with MCR^2 projection to dimension 200 achieves best accuracy without no additional time for clustering like in the same setup with k-means (denoted with *). Clustering backbone sentence embeddings from SBERT with k-means (denoted with **) took almost an hour.

model	accuracy	TIME encoding	clustering	total
all-mpnet-base-v2 + MCR50	0.562	00:04:15	-	00:04:15
all-mpnet-base-v2 + MCR100	0.545	00:04:15	-	00:04:15
all-mpnet-base-v2 + MCR200	0.645	00:04:16	-	00:04:16
all-mpnet-base-v2 + MCR300	0.632	00:04:17	-	00:04:17
all-mpnet-base-v2 + MCR50 + kmeans	0.671	00:04:15	00:07:08	00:11:23
all-mpnet-base-v2 + MCR100 + kmeans	0.650	00:04:15	00:07:22	00:11:37
all-mpnet-base-v2 + MCR200 + kmeans*	0.635	00:04:16	00:09:08	00:13:24
all-mpnet-base-v2 + MCR300 + kmeans	0.631	00:04:17	00:11:09	00:15:26
all-mpnet-base-v2 + kmeans (768)**	0.648	00:04:17	00:59:57	01:04:12
all-mpnet-base-v2 + MCR768 + kmeans	0.630	00:04:17	00:18:15	00:22:32

TABLE II: For Semantic Textual Similarity (STS) tasks backbone model achieves the best results (bolded) for Spearman rank correlation coefficient on multiple benchmarks, although we observe comparable results of our method compared to the backbone and distilled model `all-MiniLM-L6-v2`.

model	STSb	STS12	STS13	STS14	STS15	STS16
all-mpnet-base-v2 + MCR50	0.749	0.666	0.739	0.713	0.754	0.768
all-mpnet-base-v2 + MCR100	0.788	0.696	0.782	0.753	0.791	0.793
all-mpnet-base-v2 + MCR200	0.818	0.712	0.812	0.779	0.819	0.816
all-mpnet-base-v2 + MCR300	0.821	0.718	0.817	0.783	0.827	0.823
all-mpnet-base-v2 (768)	0.836	0.722	0.821	0.790	0.838	0.831
all-MiniLM-L6-v2 + MCR50	0.752	0.654	0.690	0.682	0.741	0.737
all-MiniLM-L6-v2 + MCR100	0.778	0.685	0.742	0.721	0.780	0.777
all-MiniLM-L6-v2 + MCR200	0.810	0.705	0.773	0.751	0.813	0.792
all-MiniLM-L6-v2 + MCR300	0.813	0.710	0.780	0.759	0.826	0.800
all-MiniLM-L6-v2 (384)	0.824	0.711	0.790	0.772	0.838	0.812

TABLE III: For Sentence Classification (SC) tasks backbone model achieves the best results (bolded) for accuracy on multiple benchmarks, although we observe comparable results of our method compared to the backbone and distilled model.

model	SST2	SST5	MR	CR	SUBJ	MPQA	TREC
all-mpnet-base-v2 + MCR50	75.45	36.43	69.67	63.76	79.16	68.84	51.6
all-mpnet-base-v2 + MCR100	82.54	39.19	75.85	63.76	81.86	68.77	60.0
all-mpnet-base-v2 + MCR200	86.55	42.76	80.62	72.77	88.28	82.27	71.0
all-mpnet-base-v2 + MCR300	87.59	44.66	82.33	79.71	90.73	85.76	79.8
all-mpnet-base-v2 (768)	88.74	49.00	85.05	86.84	93.97	89.32	94.0
all-MiniLM-L6-v2 + MCR50	65.95	31.76	61.61	63.76	79.19	68.77	41.0
all-MiniLM-L6-v2 + MCR100	72.27	33.94	66.38	63.82	83.97	76.58	64.0
all-MiniLM-L6-v2 + MCR200	77.54	37.10	70.06	69.17	86.87	81.83	69.2
all-MiniLM-L6-v2 + MCR300	79.35	39.50	72.95	75.07	88.47	84.13	72.0
all-MiniLM-L6-v2 (384)	81.44	42.99	75.98	80.56	91.80	87.38	90.0

The number given in parenthesis is the default embedding size.

a) Results on SR tasks: We evaluate our projection layer capacity of clustering against k-means clustering in retrieval space of sentence embeddings. The query sentence is assigned to a cluster of semantically related sentences, and we compare whether the ground truth duplicate belongs to that cluster, reported as an accuracy score. In all our experiments, the number of clusters was up to 128 (chosen empirically). In Table I and Fig. 2, accuracy scores and overall time for computation (i.e., encoding of sentence embeddings and clustering)

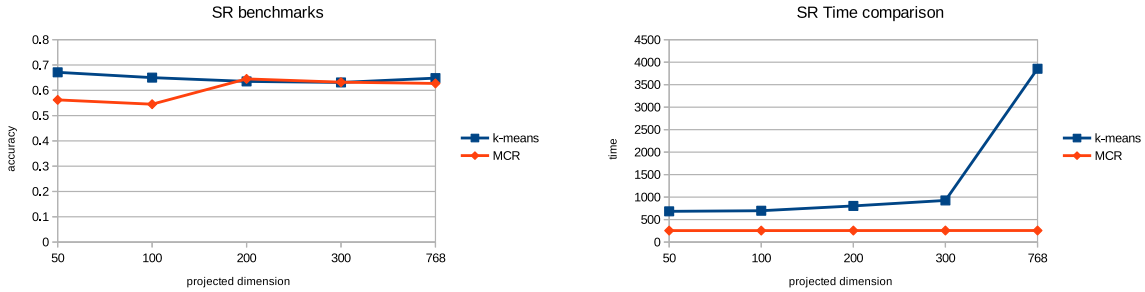

Fig. 2: Performance comparison on SR task

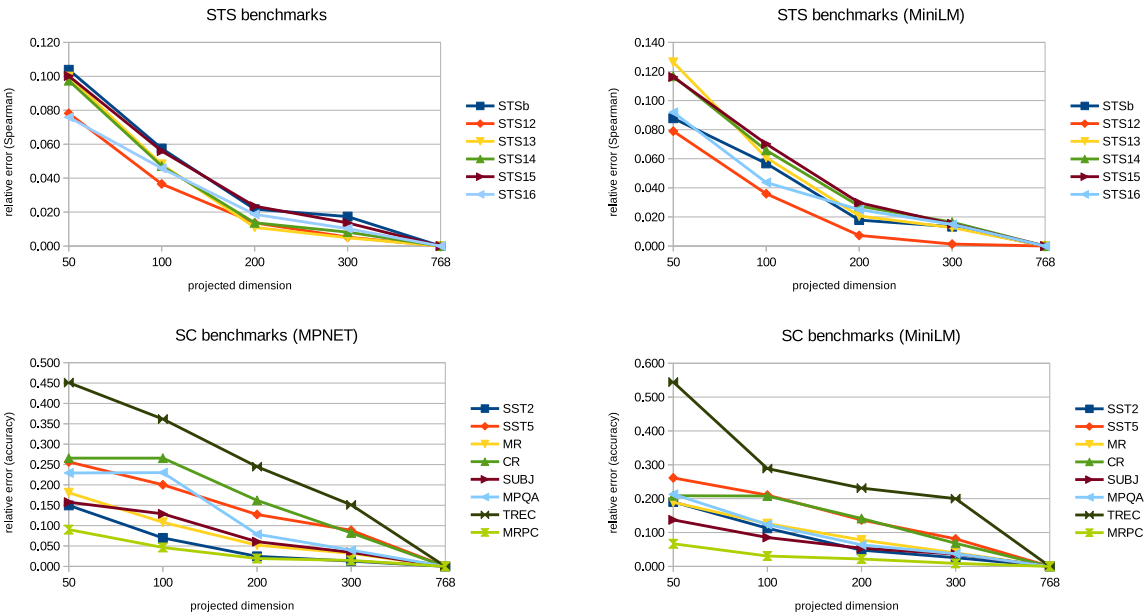

Fig. 3: Relative error in STS and SC benchmarks

depending on embedding size and type of model (MCR^2 with implicit clustering or k-means) are presented. Our method is comparable to k-means algorithm down to a certain dimension. On dimensions less than 200 k-means performed slightly better due to the fact that we did not put much effort into finding suitable λ values (suggested values of λ are from [16]). Our method computes clusters during inference which is much faster comparing to using k-means. Also worth noting, our projection layer used as a non-linear mapper in the original space (i.e., without any dimensionality reduction) retrofitted the sentence embeddings, enabling faster convergence of k-means algorithm (shown in the last row of Table I).

b) Results on STS tasks: Table II presents results for baseline (MPNET) and distilled model (MiniLM) coupled with the projection layer (MCR) with various embedding sizes. As one can see, a relative error of up to 13% in Spearman rank correlation is incurred if the sentence embedding dimension is as low as 6% of the original sentence embedding size. We conclude that due to the projection layer's ability to preserve cosine distance in lower-dimensional space, the neighborhood of points is preserved, resulting in less performance degradation. This trend is visible on all STS benchmarks with both models. The relative error in Spearman rank correlation coefficient with respect to projection dimension is shown in the first row of Fig. 3 for both the baseline and distilled model.

c) Results on SC tasks: As seen in Table III, it is observable that per-sentence classification problems like SST2 and MRPC have less performance degradation than per-token sentence classification problems like MPQA and per-sentence multi-classification problems like TREC, respectively. This is because fine-grained semantics for such tasks could not be preserved as much during projection. In the worst case, the performance degradation went up to 45% for the baseline model and up to 60% for the distilled model, respectively, at 6% of the original embedding size. The relative error in accuracy with respect to projected dimension is shown in the second row of Fig. 3 for both the baseline and distilled model.

V. Conclusion

In this paper, we demonstrated how MCR^2 technique could be used to obtain lower-dimension embeddings of

sentence representation for fast semantic retrieval tasks up to 70% of its original size. Also, we argued that these embeddings are comparable with SBERT results on standard semantic NLP benchmarks. Due to the projection layer's ability to cluster data, we were able to cluster our sentences without any extra time cost and further reduce the representation of sentences to a reasonable dimension size without significant loss of the important semantic features. We hope our approach gives new insights for possible applications in deploying AI models in smaller-scale computer systems.

REFERENCES

[1] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, no. null, p. 1137–1155, mar 2003.

[2] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," 2013. [Online]. Available: https://arxiv.org/abs/1310.4546

[3] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. [Online]. Available: https://aclanthology.org/D14-1162

[4] 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 Conference 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, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[5] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 2227–2237. [Online]. Available: https://aclanthology.org/N18-1202

[6] H. Aldarmaki and M. Diab, "Evaluation of unsupervised compositional representations," 2018. [Online]. Available: https://arxiv.org/abs/1806.04713

[7] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," Aug. 2019, arXiv:1908.10084 [cs]. [Online]. Available: http://arxiv.org/abs/1908.10084

[8] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler, "Skip-thought vectors," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'15. Cambridge, MA, USA: MIT Press, 2015, p. 3294–3302.

[9] L. Logeswaran and H. Lee, "An efficient framework for learning sentence representations," *CoRR*, vol. abs/1803.02893, 2018. [Online]. Available: http://arxiv.org/abs/1803.02893

[10] M. Pagliardini, P. Gupta, and M. Jaggi, "Unsupervised learning of sentence embeddings using compositional n-gram features," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, 2018.

[11] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *arXiv preprint arXiv:1607.04606*, 2016.

[12] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 670–680. [Online]. Available: https://aclanthology.org/D17-1070

[13] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," 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. 6894–6910. [Online]. Available: https://aclanthology.org/2021.emnlp-main.552

[14] 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. [Online]. Available: https://aclanthology.org/2020.emnlp-main.733

[15] Y. Yu, K. H. R. Chan, C. You, C. Song, and Y. Ma, "Learning Diverse and Discriminative Representations via the Principle of Maximal Coding Rate Reduction," Jun. 2020, arXiv:2006.08558 [cs, math, stat]. [Online]. Available: http://arxiv.org/abs/2006.08558

[16] Z. Li, Y. Chen, Y. LeCun, and F. T. Sommer, "Neural Manifold Clustering and Embedding," Jan. 2022, arXiv:2201.10000 [cs]. [Online]. Available: http://arxiv.org/abs/2201.10000

[17] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience, 2006.

[18] Y. Ma, H. Derksen, W. Hong, and J. Wright, "Segmentation of multivariate mixed data via lossy data coding and compression," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 9, pp. 1546–1562, 2007.

[19] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 632–642. [Online]. Available: https://aclanthology.org/D15-1075

[20] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 1112–1122. [Online]. Available: https://aclanthology.org/N18-1101

[21] I. A. M. Huijben, W. Kool, M. B. Paulus, and R. J. G. van Sloun, "A review of the gumbel-max trick and its extensions for discrete stochasticity in machine learning," 2021. [Online]. Available: https://arxiv.org/abs/2110.01515

[22] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," 2015. [Online]. Available: https://arxiv.org/abs/1511.07289

[23] D. Hoogeveen, K. M. Verspoor, and T. Baldwin, "Cqadupstack: A benchmark data set for community question-answering research," in *Proceedings of the 20th Australasian Document Computing Symposium (ADCS)*, ser. ADCS '15. New York, NY, USA: ACM, 2015, pp. 3:1–3:8. [Online]. Available: http://doi.acm.org/10.1145/2838931.2838934

[24] N. Reimers and I. Gurevych, "Making monolingual sentence embeddings multilingual using knowledge distillation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2020. [Online]. Available: https://arxiv.org/abs/2004.09813

[25] K. Patel and P. Bhattacharyya, "Towards lower bounds on number of dimensions for word embeddings," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 31–36. [Online]. Available: https://aclanthology.org/I17-2006

[26] A. Conneau and D. Kiela, "SentEval: An evaluation toolkit for universal sentence representations," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA), May 2018. [Online]. Available: https://aclanthology.org/L18-1269

[27] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.