Semantic Word and Sentence Embeddings Compression using Discrete Wavelet Transform

Rana Aref Salama[1,2], Abdou Youssef[1], and Mona Diab[3]
[1] School of Engineering and Applied Science, George Washington University, USA
[2] Faculty of Computers and Artificial Intelligence, Cairo University, Egypt
[3] Language Technologies Institute, Carnegie Mellon University, USA

Abstract

Wavelet transforms, a powerful mathematical tool, have been widely used in different domains, including Signal and Image processing, to unravel intricate patterns, enhance data representation, and extract meaningful features from data. Tangible results from their application suggest that Wavelet transforms can be applied to NLP capturing a variety of linguistic and semantic properties. In this paper, we empirically leverage the application of Discrete Wavelet Transforms (DWT) to word and sentence embeddings. We aim to showcase the capabilities of DWT in analyzing embedding representations at different levels of resolution and compressing them while maintaining their overall quality. We assess the effectiveness of DWT embeddings on semantic similarity tasks to show how DWT can be used to consolidate important semantic information in an embedding vector. We show the efficacy of the proposed paradigm using different embedding models, including large language models, on downstream tasks. Our results show that DWT can reduce the dimensionality of embeddings by 50-93% with almost no change in performance for semantic similarity tasks, while achieving superior accuracy in most downstream tasks. Our findings pave the way for applying DWT to improve NLP applications.

1 Introduction

Embedding models have evolved as a crucial part of any NLP application. Typically, they transform text, using different numerical analysis methods, into high-dimensional dense vectors that capture semantic and contextual aspects of the text for subsequent use by various tasks. As these models evolved, their complexity, dimensionality, and quality have all increased simultaneously, with a particular emphasis on quality and minimal attention to dimensionality. Typically generated embeddings are fixed in size for all tasks and are proportional to the model size, rendering their use in low resource settings a challenge. Accordingly, in this paper, we investigate the adaptation of DWT from Signal and Image processing to the field of NLP for the analysis and compression of word and sentence embeddings. DWT analyzes and reduces the size of the data by identifying redundant and important information in data. We posit that DWT has the potential to generate compressed embeddings capable of retaining contextual information and semantic relationships between words as well as among sentences. Our key contributions: 1) Introduce DWT as an effective compression method for compressing embeddings with respect to an underlying task; 2) Propose a novel approach leveraging DWT for analyzing semantics in word and sentence embeddings; 3) Study the efficacy of DWT embeddings in capturing and retaining semantics by applying them to similarity and downstream tasks using various embeddings.

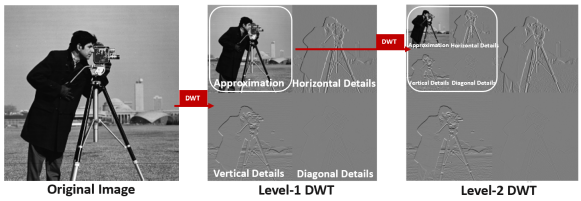

Figure 1: Applying DWT to an image. Level-1 DWT transforms the image into approximation coefficients (that resemble the original image) and vertical, horizontal and diagonal detail coefficients. Level-2 DWT can be obtained recursively from Level-1 coefficients(multi-scale analysis).

2 Motivation

The use of spectral analysis methods in Image and Signal processing is driven by their ability to analyze data in the frequency domain, providing insights and revealing hidden patterns and dynamics not detectable in the spatial domain. Methods such as the Discrete Cosine Transform (DCT) and Fourier Transform (FT) have been commonly used. However, these methods offer global frequency representations that ignore the location of frequencies in the original domain and lack the abil-

Findings of the Association for Computational Linguistics ACL 2024, pages 15963–15977

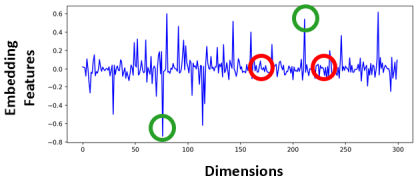

Figure 2: Graph of FastText embedding for the word 'work'(dim=300). Some features are similar in value (highlighted in red) and others exhibit significant spikes (highlighted in green).

ity to perform multi-resolution analysis. DWT effectively addresses these limitations by examining data change over time or position, i.e. frequency and time localization, and consequently captures varied information across different scales (multi-resolution analysis). As a result, DWT achieved superior results in other fields. DWT analyzes data based on their variation, filtering low-varying or highly correlated (low-frequency) and high-varying (high-frequency) components while preserving their spatial domain information. Figure1 visualizes the result of applying DWT to an image resulting in low-frequency coefficients (approximations) that capture the overall structure of the data, hence used for compression (Lewis and Knowles, 1992; Grgic et al., 2001), while high-frequency coefficients (details) capture abrupt changes and motifs, which provide information about edges and contours, making them useful for edge detection (Xizhi, 2008; Zhang et al., 2009) and noise filtering (Dautov and Özerdem, 2018). DWT can be further applied, recursively, to any coefficients of the first level to achieve a second level of the transform that contains more approximation and details coefficients. In the realm of NLP, visualizing a word embedding vector, as shown in Figure2, we observe a signal-like structure with a few spikes indicating high variation in feature values, alongside features with minimal variation. This pattern suggests that applying spectral analysis to these vectors and numerical representations can reveal additional patterns or insights that may not be immediately apparent. Few attempts for applying spectral methods to NLP including DCT (Almarwani et al., 2019) and Higher-order Dynamic Mode Decomposition (HODMD) (Kayal and Tsatsaronis, 2019) were found successful. However, we are unaware of any application of applying DWT to word and sentence embedding other than our preliminary application of DWT to enhance DCT sentence embedding in (Salama et al., 2024). In this work, we posit that DWT can effectively compress embedding representations, and analyze them at different scales to understand and capture different linguistic patterns encoded in these embeddings. This analysis allows DWT to concentrate embedding energy with high compression ratios without significant loss of information. DWT allows for more efficient representation, less storage requirements and reduced computational complexity. Additionally, DWTs are scalable and can be applied to any embedding model and for any task. Accordingly, we believe that DWT, as a spectral transform, holds promise for NLP tasks.

3 Related Work

3.1 Embedding Compression

Several studies have tackled embedding compression methods including code-book (Shu and Nakayama, 2017)(Kim et al., 2020), quantization (Ling et al., 2016)(Shi and Yu, 2018)(Tao et al., 2022) and factorization(Acharya et al., 2018). Other methods applied compression methods based on knowledge distillation (Gao et al., 2023). Some other models consider compressing model parameters (Mao et al., 2020), token embedding matrix (Bałazy et al., 2021), or prune model weights (Li et al., 2017). Yet all these techniques compress only word representations, regardless of the semantics they convey. Additionally, in the era of large language models (LLM), recent research (Wang et al., 2023) argues that the dimensionality of the embedding representations, specifically in sentence embeddings, are sub-optimal and the same encoded information may be represented in smaller dimensions while achieving comparable performance.

3.2 Spectral Methods

The success of spectral methods in studying and analyzing embeddings has recently become evident in NLP. The application of DCT in sentence embedding has shown promising results(Almarwani et al., 2019), yet it has only been applied to non-contextualized embeddings to generate sentence embeddings, not for compression. These DCT embeddings resulted in long sentence representations, corresponding to the number of coefficients considered from the transformation. We proposed using DWT to address this issue in (Salama et al., 2024), and results were promising. Similarly with HODMD (Kayal and Tsatsaronis, 2019), a sentence is represented as a signal with transitional properties captured in the frequency domain using uncorrelated coefficients to encode a sentence. Such models capture structural varia-

tion without losing on efficiency (comparable to averaging) (Zhu and de Melo, 2020) and yet outperform more complex sentence embedding models (Mikolov et al., 2018). However, these models tend to analyze frequencies along similar word embedding dimensions, on a vertical level (inter-word embedding, aka across all words), accumulating a limited number of base frequency coefficients and dropping the rest, in addition to ignoring their spatial domain position, i.e., ignoring intra-word frequencies within individual word embeddings.

4 Discrete Wavelet Transforms

DWTs are mathematical functions that analyze data, *f(t)*, using a window (a function) known as the *Mother Wavelet* (MW) $\psi(t)$, also called wavelets. This window moves over the data sequentially to examine the variations in the data with respect to this window (localized in time). Within a given analysis window if surrounding data features exhibit minimal variation, they can be grouped together and represented using Approximation coefficients (cA). Conversely, if certain features vary significantly, they are represented using Detail coefficient (cD). To shift over the data, a MW translates and dilates $\psi_{a,b}(t)$ (as in equation (1)) using a shifting factor, *b*, and scales with a scaling factor, *a*, to capture different frequency variations at different data segments in time *t*.

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi(\frac{t-b}{a}) \qquad (1)$$

The resulting coefficients are equivalent to a pair of sub-band linear (convolutional) filters (Vetterli and Kovacevic, 1996): one low-pass and one high-pass. The output of a filter pair is usually down-sampled by 2 so the combined output is of the same size (dimensionality) as the original input. This filtering + downsampling can be cascaded on multiple levels recursively to analyze data at multiple levels of resolution. Note that DWT can be compared to Convolutional Neural Networks (CNNs) (Gu et al., 2015) in that they both use sliding-window filters and downsampling. The difference is that in CNNs, the filters are learned from the training data, while in DWT the filters are designed, not learned.

There are many families of MWs: Haar, Symmlets, Coiflets, and Daubechies, to name a few (Madhavan, 2003). As a proof of concept in this paper, and in the interest of space, we will only be using and reporting on a subset of the MWs that yield the best results in our experiments. [1]

[1]For a more detailed explanation of Wavelet Transform theory, refer to (Daubechies, 1992; Madhavan, 2003; Brunton and Kutz, 2019).

5 Method

DWT filters features in embeddings by capturing how consecutive features in an embedding vector change, where small numerical difference or variation are converted into approximation coefficients. While features with large variations are converted into detail coefficients, and hence *reflecting* the structure and behavior of the underlying information encoded in the vector. As shown in Figure3, given a word or a sentence embedding E_d of a *d*-dimensional vector, we transform E_d by applying DWT, $DWT(E)$, which decomposes the embedding vector into two vectors of low-varying coefficients, cA, and high varying coefficients, cD, for one level of transformation. The dimension for each set of coefficients, d', is reduced by 2, $d' = d/2$. The generated coefficients, cA and cD, can be further transformed for a second level, generating new vectors of approximation and detail coefficients, and downsampled by 2 again. In fact, this recursive process can be repeated many times, say L times. L represents the number of DWT levels, and is determined based on the required compression ratio and performance. For the selection of the MW used in the transformation, we primarily use Symlets, Daubechies, and Coiflets wavelets across all experiments and the best performance per task is recorded. [2]

Coefficients Selection: In image processing, different sets of coefficients capture different aspects of an image. Similarly, we use different sets of coefficients as compact representations of the base embedding. We analyze the amount of information each set of coefficient encodes by studying their effectiveness in similarity and downstream tasks. We consider the following possible selection mechanisms: (a) *Level-1 Coefficients:* Employ Level-1 coefficients, either the approximation (cA) or detail (cD) coefficients as the compressed representation with a size downsampled by 2. (b) *Higher-Level Coefficients:* Employ coefficients from higher levels of the DWT (i.e., Levels 2, 3 or later). Specifically, we examine Level-2 approximation coefficients derived from Level-1 approximation and detail coefficients, namely cAA (approximation of approximation) and cDA (approximation of details) which are a quarter the size of the original

[2]In the interest of simplicity and clarity, we will omit the details of the specific MW used in each experiment, as this paper is intended as a proof of concept, and not for comparing different MWs.

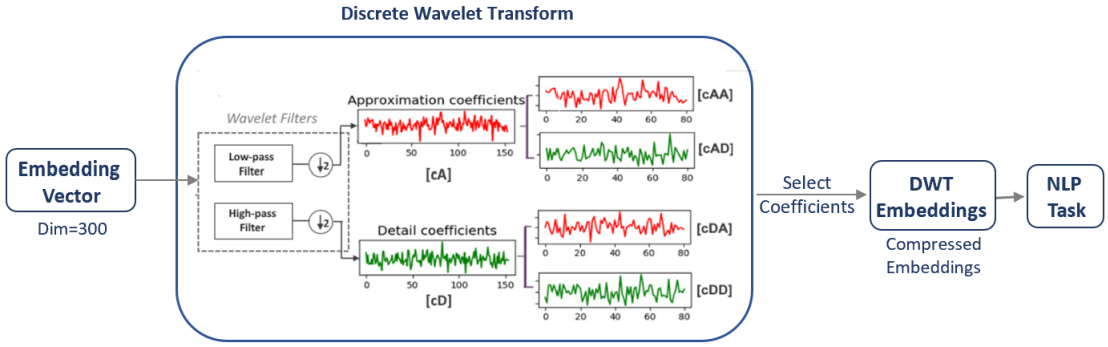

Figure 3: Transforming the embedding vector in Figure 2 using a 2-level DWT yielding cA and cD coefficients (dim=150) at Level-1. At Level-2, we get cAA and cAD from Level-1 approximations, and cDA and cDD from Level-1 details (dim=80).

embedding vector. Additionally, for a higher compression ratio, we explore approximation coefficients from subsequent Level-3 (cAAA) and Level-4 (cAAAA), Which are 1/8 and 1/16 the size of the original embedding vector, respectively. Our emphasis lies on approximation for deeper compression, as they encapsulate an abstracted representation of the original embedding. (c) *Combined Coefficients:* In experiments where Level-1 coefficients do not achieve comparable performance, we incorporate coefficients from higher levels. Specifically, if Level-1 approximation doesn't sufficiently encode all relevant semantics, we further combine it with Level-2 approximation (derived from Level-1 detail), referred to as cDA. Similarly, for detail, we combine it with Level-2 detail (derived from Level-1 approximation), referred to as cAD. As illustrated in Figure 3, we enrich the red-colored coefficients from Level-1 (approximation) with the red-colored coefficients from Level-2 (obtained from Level-1 green coefficients). Thus, the combined embeddings are cA+cDA and similarly cD+cAD. Note that combining Level-2 red-colored coefficients, cAA, with cA (Level-1 red-colored) is irrelevant since they contain redundant information.

6 Evaluation

To evaluate the proposed approach, we first investigate its efficacy in capturing and compressing semantics in semantic similarity tasks. We further evaluate their effectiveness in a number of downstream tasks to evaluate their efficacy extrinsically. We consider different embeddings, baselines, experimental setup and tasks as follows.

Embeddings: We experiment with different embeddings; Pre-trained Language Models as BERT (Devlin et al., 2018), GPT (Radford and Narasimhan, 2018), SBERT (Wang and Kuo, 2020) and RoBERTa (Liu et al., 2019) sentence embeddings. We also include non-contextualized embeddings from our previous work (Salama et al., 2024) for completeness. We use GloVe (Pennington et al., 2014) and FastText (Mikolov et al., 2018) with various dimensions (50, 100, 200, 300).

Word Embedding	Dim	SimLex	WS353	MEN
$GloVe_{100}$	100	12.22	46.96	57.73
$GloVe_{50}$	50	9.82	42.17	53.05
GloVe+DWTcD	50	11.50	**50.19**	**58.96**
GloVe+DWTcA	50	**13.48**	44.08	57.21
$GloVe_{200}$	200	13.03	48.00	59.42
$GloVe_{100}$	100	12.22	46.96	57.73
GloVe+DWTcD	100	11.50	**50.19**	58.96
Glove+DWTcA	100	**20.79**	**50.19**	**62.00**
FastText	300	50.30	**79.13**	**83.36**
PCA	150	27.01	52.22	63.31
$FastText+DWT_{cD}$	150	**50.32**	74.18	79.91
$FastTex+DWT_{cA}$	150	49.03	75.44	80.96
$FastText+DWT_{cD+cAD}$	225	**50.32**	78.34	82.59
$FastText+DWT_{cA+cDA}$	225	49.05	75.56	82.96

Table 1: Spearman Rank Order Correlation (SPC) results on SimLex-999, WS353 and MEN datasets; using GloVe-Twitter27B embeddings compared to Level-1 DWT coefficients. Baseline includes base GloVe embeddings of similar size in addition to GloVe with 50% less dimensions. DWT_{cD} and DWT_{cA} correspond to the embeddings yielded at Level-1 DWT transform. Level-2 DWT_{cD+cAD} and DWT_{cA+cDA} coefficients from Level-1 coefficients(dim=150) concatenated with Level-2 (dim=75) Best results are in bold and best results per experimental condition are in red.

Baselines: For every experiment, we use different baselines, in addition to the base embeddings, to explore the capabilities of DWT. Our baselines include: (1) Base embeddings (in all experiments); (2) Other dimensionality reduction methods to assess their effectiveness in comparison to DWT: PCA for dimensionality reduction(Shlens, 2014) and DCT for compression(Gupta and Garg, 2012) in some of our experiments.

Experimental Setup: For all sentence embeddings experiments, we use the SentEval toolkit (Conneau and Kiela, 2018) for evaluation. For all down-

	Dim	SimLex	WS353	MEN
BERTbase	768	60.75	28.00	**59.55**
BERT+DWTcD	383	60.31	**28.41**	59.19
BERT+DWTcA	383	**60.90**	28.25	59.31
BERTLarge	1024	69.72	44.00	62.18
BERT+DWTcD	512	69.65	**45.05**	**62.68**
BERT+DWTcA	512	**69.95**	43.31	61.09
GPTBase	1536	**50.00**	**64.89**	73.00
GPT+DWTcD	768	49.95	63.88	**73.34**
GPT+DWTcA	768	49.37	64.58	71.82
GPTLarge	3072	56.60	72.31	78.35
GPT+DWTcD	1535	56.23	72.03	77.57
GPT+DWTcA	1535	**56.93**	71.93	77.29

Table 2: Similar experiment as Table 1 using BERT and GPT base and large word embeddings with base embeddings as the baseline.

stream tasks, we leverage multi-layer perceptron (MLP) classifiers based on the default setup outlined in SentEval.[3]

6.1 Semantic Similarity Evaluation

Word and sentence similarity tasks have become the de-facto method for semantic evaluation (Wang et al., 2022). Semantic Similarity involves measuring the degree of relatedness and similarity between pairs of words or sentences compared against human judgments or similarity scores assigned by human annotators.

6.1.1 Word Similarity Evaluation

We start by evaluating DWT embeddings for word similarity, using the following datasets: SimLex-999 (Hill et al., 2014), MEN (Bruni et al., 2014) and WS353 (Finkelstein et al., 2001).

In our initial experiment, we utilize DWT Level-1 approximation (cA) and detail (cD) coefficients as DWT embeddings for the word semantic similarity task. In this experiment we assess: (1) GloVe embeddings with dimensions 100 and 200. Baselines are: the base GloVe embeddings of the same size, alongside the base GloVe embeddings originally reduced in dimensions by 50%. (2) FastText embeddings with dimension 300. Baselines are: the base FastText embeddings, and PCA reduced embedding with size 150 dimensions. (3) BERT and GPT models, where for each model we consider two variants, base and large models, in order to conduct a thorough evaluation with original embeddings as baselines. We found DWT embeddings to consistently outperform the baselines for GloVe embeddings, with dimensions 100 and 200, as depicted in Table 1. DWT embeddings efficiently compress the semantics encoded in the original embeddings, surpassing both baselines with a 50% reduction in dimensionality and in some cases surpassing the performance of embeddings that are four times larger as in SimLex and WS353, where DWT embeddings of size 50 outperforms GloVe embeddings of size 200. This empirically shows that DWT embeddings not only serve as a dimensionality reduction technique but also adeptly capture the semantics encoded in an embedding vector and effectively compress them in DWT coefficients. Additionally, DWT embeddings significantly outperform PCA-reduced embeddings, demonstrating superior performance with >20% improvement. Additionally, in the case of contextualized embeddings, DWT embeddings exhibit comparable performance to all baselines for all models. This suggests that the large dimensionality used for these models may not be significant for encoding words. It can also be noted that the DWT BERT embeddings are more comparable to the baseline than DWT GPT embeddings; this suggests that GPT models are packing more information and semantics than the BERT model for these datasets. This empirically shows that DWT reveals new aspects of embeddings that were not evident in the original domain. Nevertheless, for FastText, while DWT yields similar performance to the original embeddings for the SimLex dataset, yet for WS353 and MEN datasets Level-1 DWT embeddings failed to fully capture the encoded semantics present in the original embeddings. This discrepancy suggests that FastText embeddings contain richer semantics compared to GloVe embeddings, a conclusion supported by the similarity results achieved with the original embeddings.

We further use the combined coefficients; cA+cDA and cA+cDA. As shown in Table 1, the combined DWT embeddings at a compression of 25% dimensionality augmenting the embeddings with additional semantics, resulting in a performance comparable to the baseline with a slight reduction approx. 1% in performance for WS353 and MEN datasets. Conversely, for SimLex datasets, the addition of coefficients from subsequent layers (cD+cAD) did not improve the performance and cD coefficients seems to have effectively encoded the semantics for all word embeddings. Nevertheless, it is crucial to emphasize that the selection of DWT coefficients depends not only on the type of embeddings but also on the particular task in consideration. As a result, we utilize DWT embeddings generated from

[3]The source code is publicly available on GitHub at https://github.com/engranas/DWT-Semantic-Compression

Word	BERT(dim=768)	BERT+DWT$_{cA}$(dim=383)	BERT+DWT$_{cD}$(dim=383)
happy	happy, sad, pleased, smiling, thrilled	happy, smiling, excited, pleased, glad	happy, sad, joy, pleased, thrilled
sea	sea, gulf, marine, desert, underwater	sea, marine, desert, gulf, island	sea, gulf, underwater, beach, fish
playing	playing, riding, practicing, throwing, creating	playing, creating, riding, writing, coloring	playing, riding, fighting, throwing, plays

Table 3: 5-nearest cosine similar words using BERT embeddings.

	Dim	AP	BM	BLESS
FastText	300	**0.70**	**0.47**	**0.86**
FastText+DWT$_{cD}$	150	**0.70**	0.46	**0.87**
FastText+DWT$_{cA}$	150	**0.70**	**0.49**	0.82

Table 4: Results for Concept Categorization task for 3 standard datasets: AP, BM and BLESS using using Level-1 cA and cD coefficients.

FastText for another semantic task, Concept Categorization, to further elaborate on the adaptability of DWT for different tasks. Concept Categorization groups words in different categories based on semantic clusters (Baroni et al., 2014). For this evaluation we use the datasets: AP (Almuhareb, 2006), BM (Murphy et al., 2012) and BLESS datasets (Baroni and Lenci, 2011). We use the base embeddings as the baseline. As illustrated in Table 4, cA and cD embeddings demonstrate comparable or even superior results compared to the baseline, despite 50% dimensionality reduction. This illustrates the effectiveness of DWT embeddings in encapsulating essential semantics from the original embedding for the given task. To conduct a qualitative analysis of DWT embeddings, Table 3 displays the five nearest neighbors (determined by cosine similarity) for randomly chosen words utilizing BERT word embeddings. As shown, approximation coefficients capture more relevant words like 'excited' and 'glad' for the word 'happy'. Also, 'island' for 'sea', and 'writing' and 'coloring' for the word 'playing'. Detail coefficients capture more appropriate words such as 'joy' for 'happy', 'beach' and 'fish' for 'sea', and 'fighting' for 'playing'. Additionally, it's apparent that certain relevant words, which share similar meanings or contexts, have closer similarity, such as "creating" for "playing" and "smiling" for "happy".

6.1.2 Sentence Semantic Similarity

The Semantic Textual Similarity (STS) Task is a common benchmark used for evaluating the performance of semantic models. In this setting, we examine the application of DWT to contextualized sentence embeddings on STS tasks 2012 and 2016 (Agirre et al., 2012, 2016), STS benchmark (STSB) (Cer et al., 2017) and SICK-Relatedness (Marelli et al., 2014). (See Appendix for more STS tasks results.) We evaluate DWT embeddings on contextualized models as they are becoming dominant for sentence embedding (Wang et al., 2024). We consider SBERT base and large models (Wang and Kuo, 2020), RoBERTa base and large models (Liu et al., 2019).[4] In this experiment, we will further explore the effectiveness of DWT in capturing relevant features and semantics within pretrained language models, illustrating that the extended size of dimensionality of these embeddings is not optimal for semantic representation, which we found consistent with the recent results and findings in (Wang et al., 2023) without the need for any further training or fine-tuning of the model. We consider Level-1 coefficients, cA and cD, with 50% reduction in dimensions. We also consider Level-2 coefficients, cAA and cDA, with 75% reduction in dimension. For the baselines we consider: (1) The base embeddings, (2) DCT (Gupta and Garg, 2012) as another spectral model used for lossy compression. We use DCT to compress the original SBERT and RoBERTa embeddings by applying the DCT transform to these vectors and selecting the first n coefficients, where n is equivalent to 50% or 25% of the original embedding size.[5] As shown in Table 5, DWT consistently surpasses DCT across all tasks, particularly in STSB and SICKR, where the efficacy of DCT embeddings reflects significantly lower performance. Although DCT was capable of compressing embeddings and maintaining comparable performance in STS12 and STS16, DCT compresses by discarding residual frequency coefficients leading to the loss of important details and a subsequent decline in performance in STSB and SICKR. DWT performs similarly to the base embeddings, with at most a 0.5% decrease in performance on a few tasks for all models using Level-1 coefficients in SBERT and RoBERTa. This indicates that Level-1 coefficients effectively capture relevant semantics, condensing them into fewer dimensions. Notably, for SBERT, the approximation coefficients yield better performance, while for RoBERTa, the detail coefficients perform

[4]Models available in https://huggingface.co/sentence-transformers; SBERT-base-nli-v2, SBERT-Large-nli-v2, nli-roberta-base and nli-roberta-large.

[5]In this context, DCT is not applied in the same way as proposed by (Almarwani et al., 2019), where the transform is applied across all words in a sentence to encode the sentence. In our context we apply it within a word vector.

Model	Dim	STS12	STS16	STSB	SICKR	Model	Dim	STS12	STS16	STSB	SICKR
$SBERT_{Base}$	768	74.09	84.08	85.35	**80.69**	$RoBERTa_{Base}$	768	69.02	83.39	81.89	80.57
SBERT+DCT	384	74.14	84.2	75.57	53.05	RoBERTa+DCT	384	68.64	82.87	65.01	39.14
SBERT+DCT	192	72.69	83.40	74.99	51.55	RoBERTa+DCT	192	67.97	82.53	67.14	36.30
$SBERT+DWT_{cD}$	384	73.57	83.83	85.83	80.07	$RoBERTa+DWT_{cD}$	384	**69.32**	**83.57**	82.40	**80.79**
$SBERT+DWT_{cA}$	384	**74.26**	**84.27**	**85.98**	80.55	$RoBERTa+DWT_{cA}$	384	68.88	82.93	82.48	80.71
$SBERT+DWT_{cAA}$	192	73.72	83.9	85.77	80.02	$RoBERTa+DWT_{cAA}$	192	68.17	82.63	82.63	80.71
$SBERT+DWT_{cDA}$	192	73.21	83.90	85.70	79.56	$RoBERTa+DWT_{cDA}$	192	68.63	82.91	**83.60**	80.31
$SBERT_{Large}$	1024	67.97	81.69	78.26	80.24	$RoBERTa_{Large}$	1024	65.07	75.70	74.68	79.86
SBERT+DCT	512	67.83	75.66	62.99	41.92	RoBERTa+DCT	512	64.79	75.07	44.00	37.23
SBERT+DCT	256	65.41	73.65	61.55	40.25	RoBERTa+DCT	256	63.05	74.12	30.52	28.23
$SBERT+DWT_{cD}$	512	67.83	81.49	80.97	**80.95**	$RoBERTa+DWT_{cD}$	512	**65.07**	**75.70**	76.76	79.65
$SBERT+DWT_{cA}$	512	**67.97**	**81.69**	80.15	80.44	$RoBERTa+DWT_{cA}$	512	64.97	75.67	75.72	79.68
$SBERT+DWT_{cAA}$	256	67.96	81.34	**82.42**	79.69	$RoBERTa+DWT_{cAA}$	256	64.94	75.5	**77.14**	78.71
$SBERT+DWT_{cDA}$	256	67.57	81.15	82.33	79.94	$RoBERTa+DWT_{cDA}$	256	64.87	75.14	**77.14**	79.03

Table 5: Results on the STS benchmark, Spearman's correlation is reported. Baseline represents the original and DCT reduced embeddings for SBERT and RoBERTa models. The best overall results are shown in bold. Best results per condition are shown in red.

better. This empirically demonstrates that SBERT and RoBERTa capture different types of features: SBERT captures more relational semantic features, whereas RoBERTa focuses on distinct semantic features. Additionally, DWT reveals some characteristics of these embeddings: large models tend to contain more redundant features. As a result, DWT can effectively compress them to 75% less dimensions while maintaining comparable performance across all tasks. More interestingly, for the STSB task using SBERT and RoBERTa, DWT outperforms the baseline in a manner proportional to the model size. Specifically, in the large models, cAA coefficients improve performance by 4% in SBERT and by 2.5% in RoBERTa. This shows that the cAA coefficients contain more dense semantic information suggesting that the original embeddings had highly correlated dimensions with redundant features that proved to be irrelevant when filtered out using DWT. Generally, DWT embeddings prove to have comparable performance or better in most tasks, at 50% reduction in embedding size. At a 75% reduction in size, the performance is still comparable and never drops more than 2%, and for some tasks the performance is significantly better. In most cases cA represents a good approximation for the original representation.

6.2 Downstream Tasks

To further evaluate our model extrinsically, we explore applying DWT embeddings to the following downstream tasks: sentiment classification on Movie Reviews (MR), Stanford Sentiment Treebank (SST2, SST5) (Pang and Lee, 2004a), product review (CR) (Hu and Liu, 2004), subjectivity classification (SUBJ) (Pang and Lee, 2004b), opinion polarity classification (MPQA), question type classification (TREC) (Voorhees and Tice, 2000), paraphrase identification (MRPC) (Dolan et al., 2004), and entailment classification on the SICK dataset (SICK-E) (Bouden and Nibouche, 2012). Due space limitation, we will only consider RoBERTa base and large embeddings for these experiments as a proof of concept, to show the efficacy of DWT. Table 6 shows the results with base embeddings as the baseline. As shown, DWT effectively compresses the base embeddings, outperforming the baselines in all tasks. Level-1 coefficients, cD, achieve better performance by 4.5% in SST5 task, 2% in TREC task, while cA outperforms in MR and SICK-E. Using Level-2 coefficients, cAA, further improves over the baseline in MPQA and MRPC, while cDA outperforms in SST2. Generally, DWT embedding surpassed the baselines with all coefficients for all tasks except for CR and TREC where the Level-2 coefficients is comparable. This demonstrates the efficacy of DWT embeddings for downstream tasks. In conclusion, we find that DWT presents an effective balance between efficiency (compactness) and accuracy, serving as an efficient data-size reduction method that condenses embeddings with relevant features, leading to enhanced performance.

Multiple Levels of DWT We further investigate the effectiveness of applying multiple levels of DWT transformation to RoBERTa large embeddings. We extend our analysis up to 4 levels of transformation, thereby achieving compression of up to 93%. We apply DWT recursively to the approximation coefficients from each level, resulting in coefficients of sizes 512, 256, 128, and 64 respectively. The results[6]in Figure 4 show that for dimensions of size 128 and 64, corresponding to Level-3 and Level-4 coefficients, with a reduction in dimensionality of 87%-94%, respectively. DWT embeddings outperform the baseline for all tasks by

[6]Detailed results for this experiment can be found in the Appendix.

		Sentiment Analysis					Inference	Paraphrase	SUBJ	TREC
Embedding	**Dim**	**MR**	**CR**	**SST2**	**SST5**	**MPQA**	**SICK-E**	**MRPC**		
$\text{RoBERTa}_{\text{Base}}$	768	85.32	90.46	92.31	52.17	89.17	80.01	70.72	**94.71**	**92.40**
DWT_{cA}	384	85.28	91.20	91.82	53.57	**89.70**	79.16	73.62	94.21	90.60
DWT_{cD}	384	**85.34**	90.97	**92.53**	**53.71**	89.38	79.58	72.23	94.38	90.41
DWT_{cAA}	192	84.59	**91.36**	90.94	53.03	89.18	**80.31**	**73.97**	93.23	84.82
DWT_{cDA}	192	84.78	91.13	91.43	52.94	89.27	79.93	73.57	93.32	82.60
$\text{RoBERTa}_{\text{Large}}$	1024	85.01	91.18	91.38	50.95	90.13	80.68	76.17	92.00	85.84
DWT_{cA}	512	**85.97**	91.26	91.60	52.22	90.62	**82.04**	77.22	92.21	87.20
DWT_{cD}	512	85.57	**91.44**	91.71	**55.34**	90.48	81.63	77.16	**92.43**	**87.60**
DWT_{cAA}	256	85.39	91.29	91.65	53.85	**90.72**	81.33	**77.39**	91.80	85.20
DWT_{cDA}	256	85.40	91.07	**91.93**	53.39	90.51	80.96	76.93	92.08	85.80

Table 6: Best Classification accuracy results on various classification tasks for Level-1 approxmation and details coefficients; cA and cD, and Level-2 DWT coefficients; cAA and cDA. The Baseline is the original RoBERTa for Base and Large models. The best overall results are shown in bold. Best results per condition are shown in red.

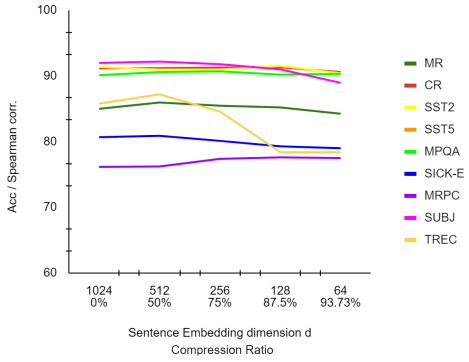

Figure 4: Accuracy results for 4-Levels of DWT for approximation coefficients for downstream tasks using RoBERTa Large embeddings

0.09-1.5% except for SICK-E, SUBJ, and TREC, which experience degradation by 1.3%, 1%, and 7.4%, respectively, although still outperforming the baseline in higher levels. This decline in performance tends to correlate with an increased compression ratio indicating that the minimized dimension size is not sufficient to fully encode the necessary semantic information, resulting in a tolerable performance decline. On the other hand, other tasks like MRPC outperform the baseline by 1.5% with just 64 dimensions, indicating that DWT effectively consolidates relevant features into much fewer dimensions, while the extended dimension size contains redundant features. These findings align with those of (Wang et al., 2023), where their model, despite being trained using a two-step method, consistently degrades performance for all tasks by 2%-9% in dimension sizes of 64. In contrast, our model exhibits comparability to the baseline in most tasks and experiences degradation by 2%-3.5% in MR, SICK-E, and SUBJ, and by 7.4% in TREC.

7 Conclusion and Discussion

In this paper, we explored the effectiveness of applying DWT to word and sentence embeddings to *selectively* reduce embeddings. Our experiments illustrate the potential of DWT to enhance space and computational efficiency without decline in performance, reducing embedding size by 50-75%. DWT reveals new insights about embeddings, and exposes hidden patterns and semantic information that were not apparent in the base embedding space. Additionally, the generated DWT embeddings postulate that different sets of coefficients capture different semantic aspects of an embedding. We conclude our study outcomes in the following points:

Correlation and Redundancy: While it has not been proven that features within the same embedding are correlated, the comparable performance achieved by DWT suggests a correlation between the dimensions of an embedding. This challenges the previous assumption that these dimensions are uncorrelated. If no correlation existed, DWT would hardly achieve comparable results to the performance of base embeddings. Our results also prove that embeddings from complex models, such as Pre-trained Language Models, contain redundant features and hence compressing them with 75% reduction in dimensionality achieves a comparable performance to the base embeddings with no more than 2% degradation.

Dimensionality Reduction Techniques: Although the general idea of DWT appears to be similar to other dimensionality reduction methods in the context of decomposition and compression, its overall properties and methodology are different (Raunak et al., 2019). DWT outperforms other compression techniques such as PCA and DCT. DWT compresses the data using localized sub-band frequency components capturing both low and high features-variation. DWT stands out as a method that can be applied universally to diverse datasets and models. PCA reduces data to a low dimension space, and de-correlates the data, but is much slower because it needs to compute the eigenvec-

tors of the original data (Priya, 2014). As for DCT, despite being a frequency analysis method, it tend to capture important frequencies indiscriminately, lacking localization and discarding other frequencies and accordingly degrading performance.

Mother Wavelets: MW is a component of DWT that controls feature filtering, with each MW corresponding to a unique pair of filters. We did not explore specific MW applications for each task due to space constraints, but limited ourselves to a predetermined number of MW families such that we maintain experimental consistency while empirically investigating the effectiveness of DWT. Our experiments indicate that Coiflets wavelets generally perform well across tasks. It is important to note that MW filtering is controlled by a scaling factor, which adjusts the level of variation considered in an embedding. For example, a Coiflet MW with scale 4 focuses on nearby features and their fine variations, whereas scale 17 emphasizes broader approximations, omitting fine details. Since the choice of the best MW varies from base embedding to base embedding, the selection and optimization of MW requires further study beyond this paper's scope. As a result, we leave further exploration of MW details and selection for future research.

Coefficients Selection: Much as in image processing, different coefficients tend to capture distinct aspects of the data. Therefore, it is essential to select the set of coefficients that are most suitable for the base embedding being used in a given task. Some tasks may benefit more from approximations over details, while others may see improved performance with nuanced information, and yet others may use both equally. However, it can be concluded that if minimizing the embedding size is a significant consideration, utilizing only Level-1 coefficients (cA or cD) with a 50% reduction in size results in comparable performance to the baselines across nearly all tasks. Moreover, approximation coefficients consistently maintain comparable performance across multiple levels of DWT, as demonstrated in the 4-level analysis, with the exception of TREC. This consistency suggests that these coefficients retain relevant information about the original embeddings. Nevertheless, a detailed study and analysis of coefficient selection fall beyond the scope of this paper and are planned for future work.

DWT for Compression: DWT offers a powerful method for compressing embeddings by exploiting correlations and reducing redundancy. The performance of DWT in compressing embeddings strongly suggests that approximation coefficients can be effectively utilized for compression, akin to their role in image compression, where they typically capture essential features of the data. In the compression process, approximation coefficients not only reduce the embedding dimensionality but also reduce noise while retaining relevant features. By reducing noise, the performance of the model improves as it casts down on the error rate thus improving the accuracy of these results. Selecting the appropriate compression level for a given task is a hyperparameter that depends on various factors, including space complexity, resources allocated for a particular task and the trade-off between effectiveness and compression ratio.

Computational Complexity: The DWT transformation is implemented through convolution and down-sampling (filtering) operations, which are typically linear with respect to the size of the input data. The overall computational complexity is often expressed in terms of the dimension of the embedding, d, so for a single level of transformation, the complexity is $O(d)$. However, the recursive nature of multi-level transformations can lead to increased computations, depending on the number of levels, L, resulting in an overall complexity of $O(Ld)$. Yet, DWT facilitates efficient dimensionality reduction, significantly reducing memory usage and increasing throughput by shrinking the size of embeddings while preserving essential features.

DWT Efficacy: Based on our evaluations, the application of DWT in the context of NLP exhibits significant potential for efficiently modeling word and sentence embeddings. Our findings demonstrate that DWT coefficients have the ability to capture various aspects of the data, with the approximation coefficients serving as a general approximation of an embedding, akin to their behavior in image analysis. Furthermore, the details coefficients excel at capturing semantic nuances within the embeddings. Notably, DWT embeddings reveal new aspects and characteristics that were previously unexplored, such as establishing correlations between dimensions of a word embedding. Additionally, DWT can adapt to any embedding: words, sentences or documents of any length in the same manner as mentioned in this paper. Overall, the application of DWT holds promise, and we anticipate its effectiveness in other NLP applications.

8 Limitations

In this paper, our focus is to thoroughly and empirically investigate the effectiveness of applying Discrete Wavelet Transform (DWT) to word and sentence embeddings, with a primary emphasis on analysis and compression. However, DWT transformations are mainly based on the selection of Mother Wavelets (MWs), which we did not specify for every experiment due to space constraints. We employed Coiflets, Symlets, Haar, and Daubechies as MWs, with the recorded best overall results. Choosing the optimal MW for DWT is a common research question in Image and Signal processing, and we defer the investigation into the selection of the best MW to future work. The results presented in this paper serve as a proof of concept, indicating the potential for other MWs to further enhance performance. Additionally, our results were generated using publicly available models that undergo regular updates; thus, discrepancies may exist between our results and those published by the model owners.

9 Ethical Considerations

As we propose a novel method for applying DWT to NLP embedding, this section is divided into the following two parts.

9.1 Dataset

Intellectual Properties and Privacy Rights We make use of publicly-available data for all experiments with no modification or update for fair comparison.

9.2 NLP Application

Code Availability When used as intended, applying the models described in this paper can save people much time. Our code will be publically available to ensure reproduciblity of results

Code Reusability To run our experiments, we sometimes used public Github repositories as intended by their authors, without any modification. All such code was appropriately referenced.

Environmental Cost The experiments described in the paper makes no use of GPUs. We used CPU for our experiments. The experiments run in several hours. Several dozen experiments were run due to parameter search, and future work should experiment with distilled models for more light-weight training. We note that while our work required extensive experiments to draw sound conclusions, future work will be able to draw on these insights and need not run as many large-scale comparisons. Models in production may be trained once for use using the most promising settings.

References

Anish Acharya, Rahul Goel, Angeliki Metallinou, and Inderjit S. Dhillon. 2018. Online embedding compression for text classification using low rank matrix factorization. *CoRR*, abs/1811.00641.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. SemEval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 252–263, Denver, Colorado. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. pages 497–511.

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 32–43, Atlanta, Georgia, USA. Association for Computational Linguistics.

Nada Almarwani, Hanan Aldarmaki, and Mona Diab. 2019. Efficient sentence embedding using discrete cosine transform. 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), pages 3672–3678, Hong Kong, China. Association for Computational Linguistics.

Abdulrahman Almuhareb. 2006. Attributes in lexical acquisition.

Klaudia Bałazy, Mohammadreza Banaei, Ré mi Lebret, Jacek Tabor, and Karl Aberer. 2021. Direction is what you need: Improving word embedding compression in large language models. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*. Association for Computational Linguistics.

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

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10, Edinburgh, UK. Association for Computational Linguistics.

Toufik Bouden and Mokhtar Nibouche. 2012. *The Wavelet Transform for Image Processing Applications*.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Int. Res.*, 49(1):1–47.

Steven L. Brunton and J. Nathan Kutz. 2019. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Alexis Conneau and Douwe Kiela. 2018. Senteval: An evaluation toolkit for universal sentence representations. *CoRR*, abs/1803.05449.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Ingrid Daubechies. 1992. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, USA.

Ç. P. Dautov and M. S. Özerdem. 2018. Wavelet transform and signal denoising using wavelet method. In *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 350–356, Geneva, Switzerland. COLING.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. volume 20, pages 406–414.

Chaochen Gao, Xing Wu, Peng Wang, Jue Wang, Liangjun Zang, Zhongyuan Wang, and Songlin Hu. 2023. Distilcse: Effective knowledge distillation for contrastive sentence embeddings.

S. Grgic, M. Grgic, and B. Zovko-Cihlar. 2001. Performance analysis of image compression using wavelets. *IEEE Transactions on Industrial Electronics*, 48(3):682–695.

Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, and Gang Wang. 2015. Recent advances in convolutional neural networks. *CoRR*, abs/1512.07108.

Maneesha Gupta and Dr. Amit Kumar Garg. 2012. Analysis of image compression algorithm using dct.

Felix Hill, Roi Reichart, and Anna Korhonen. 2014. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *CoRR*, abs/1408.3456.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD '04*.

Subhradeep Kayal and George Tsatsaronis. 2019. EigenSent: Spectral sentence embeddings using higher-order dynamic mode decomposition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4536–4546, Florence, Italy. Association for Computational Linguistics.

Yeachan Kim, Kang-Min Kim, and SangKeun Lee. 2020. Adaptive compression of word embeddings. In *Annual Meeting of the Association for Computational Linguistics*.

A. S. Lewis and G. Knowles. 1992. Image compression using the 2-d wavelet transform. *IEEE Transactions on Image Processing*, 1(2):244–250.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. 2017. Pruning filters for efficient convnets.

Shaoshi Ling, Yangqiu Song, and Dan Roth. 2016. Word embeddings with limited memory. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 387–392, Berlin, Germany. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

G. Madhavan. 2003. The illustrated wavelet transform handbook - introductory theory and applications in science, engineering, medicine and finance [book review]. *IEEE Engineering in Medicine and Biology Magazine*, 22(1):92–93.

Yihuan Mao, Yujing Wang, Chufan Wu, Chen Zhang, Yang Wang, Quanlu Zhang, Yaming Yang, Yunhai Tong, and Jing Bai. 2020. LadaBERT: Lightweight adaptation of BERT through hybrid model compression. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3225–3234, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 216–223, Reykjavik, Iceland. European Language Resources Association (ELRA).

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Brian Murphy, Partha Talukdar, and Tom Mitchell. 2012. Selecting corpus-semantic models for neurolinguistic decoding. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, page 114–123, USA. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2004a. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Bo Pang and Lillian Lee. 2004b. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Kamatchi Priya. 2014. A review on linear and non-linear dimensionality reduction techniques. *Machine Learning and Applications: An International Journal*.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Vikas Raunak, Vivek Gupta, and Florian Metze. 2019. Effective dimensionality reduction for word embeddings. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 235–243, Florence, Italy. Association for Computational Linguistics.

Rana Salama, Abdou Youssef, and Mona Diab. 2024. Combining discrete wavelet and cosine transforms for efficient sentence embedding. 5th International Conference on Advanced Natural Language Processing (AdNLP 2024), Vancouver, Canada.

Kaiyu Shi and Kai Yu. 2018. Structured Word Embedding for Low Memory Neural Network Language Model. In *Proc. Interspeech 2018*, pages 1254–1258.

Jonathon Shlens. 2014. A tutorial on principal component analysis. *Educational*, 51.

Raphael Shu and Hideki Nakayama. 2017. Compressing word embeddings via deep compositional code learning. *CoRR*, abs/1711.01068.

Chaofan Tao, Lu Hou, Wei Zhang, Lifeng Shang, Xin Jiang, Qun Liu, Ping Luo, and Ngai Wong. 2022. Compression of generative pre-trained language models via quantization.

Martin Vetterli and Jelena Kovacevic. 1996. Wavelets and subband coding. *Journal of Electronic Imaging*.

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 200–207, New York, NY, USA. Association for Computing Machinery.

Bin Wang and C. C. Jay Kuo. 2020. Sbert-wk: A sentence embedding method by dissecting bert-based word models.

Bin Wang, C. C. Jay Kuo, and Haizhou Li. 2022. Just rank: Rethinking evaluation with word and sentence similarities.

Hongwei Wang, Hongming Zhang, and Dong Yu. 2023. On the dimensionality of sentence embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10344–10354, Singapore. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models.

Z. Xizhi. 2008. The application of wavelet transform in digital image processing. In *2008 International Conference on MultiMedia and Information Technology*, pages 326–329.

Zhen Zhang, Siliang Ma, Hui Liu, and Yuexin Gong. 2009. An edge detection approach based on directional wavelet transform. *Computers Mathematics with Applications*, 57(8):1265 – 1271.

Xunjie Zhu and Gerard de Melo. 2020. Sentence analogies: Exploring linguistic relationships and regularities in sentence embeddings.

A Appendix

A.1 Sentence Semantic Similarity

Visualizing Embeddings of Words in a Sentence By sketching out all the words in a sentence like "it's a hot and sunny day," as shown in Figure 5, we observe that the energy of all word embeddings overlaps within an average sub-band. This provides a compelling explanation for why word averaging effectively represents a sentence, offering empirical support for averaging as a method for sentence embedding. This suggests that further spectral analysis of these embedding representations is promising and likely to achieve effective results, similar to those in image and signal processing.

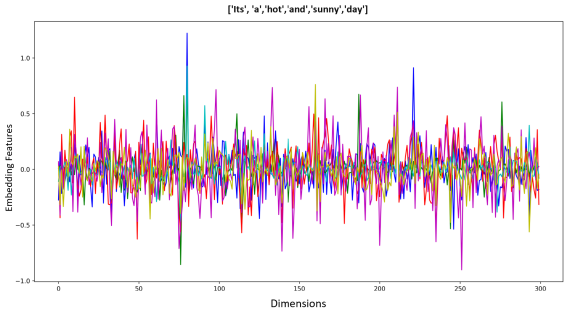

Figure 5: Similarity Matrix between 2 sentences using BERT embeddings of dimension 768, and Level-1 cA and cD of the transformed BERT embedding with dimension of 384.

Qualitative Analysis To further explore the efficacy of DWT through Sentence Similarity Tasks, we initially demonstrates the word similarity matrix between two randomly selected sentences from the STSB dataset using BERT(Devlin et al., 2018) embeddings with a dimension size of 768 as opposed to their Level-1 DWT cA and cD coefficients, with a dimension size down-sampled by 2, i.e. dimension size is 384. By comparing the overall distribution of similarity shown in Figure 6, we observe that it remains largely coherent between the original embeddings and the DWT embeddings with 50% reduction in dimensions.

Extended Intrinsic Evaluation:

1- InferSent Embeddings We additionally consider InferSent sentence embedding model(Conneau et al., 2017) for this experiment as an example for a parameterized sentence embedding model. We set the original embeddings as the baseline. As shown in Table7, Level-1 cD

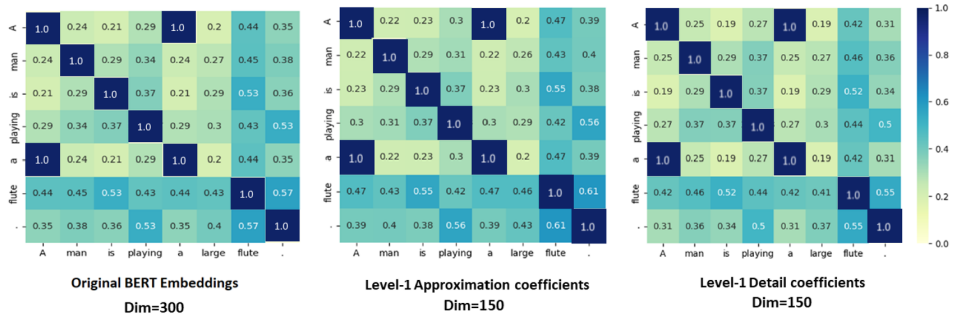

Figure 6: Similarity Matrix between 2 sentences using BERT embeddings of dimension 768, and Level-1 cA and cD of the transformed BERT embedding with dimension of 384.

Model		Dim	STS12	STS13	STS14	STS15	STS16	STSB	SICKR
InferSent	Baseline	4096	50.05	45.64	57.40	62.21	59.44	**67.19**	**81.95**
	DWT_{cD}	2053	52.93	**48.05**	**58.23**	63.90	61.63	66.27	81.06
	DWT_{cA}	2053	48.16	44.32	56.13	60.78	58.04	64.91	80.57
	DWT_{cAA}	1030	46.09	42.64	54.36	59.06	56.83	61.55	78.76
	DWT_{cDA}	1030	**53.10**	47.87	58.13	**63.94**	**61.80**	65.73	79.31

Table 7: Results on the STS benchmark, Spearman's correlation is reported. Baseline represents the original embedding and corresponding performance for InferSent model. The best overall results are shown in bold. Best results per condition are shown in red.

Model		Dim	STS13	STS14	STS15
$SBERT_{Base}$	Baseline	768	83.56	**90.73**	**88.03**
	SBERT+DWT_{cD}	384	83.06	90.55	87.56
	SBERT+DWT_{cA}	384	**83.60**	90.44	87.77
	SBERT+DWT_{cAA}	192	82.05	90.11	86.99
	SBERT+DWT_{cDA}	192	81.58	90.11	86.40
$SBERT_{Large}$	Baseline	1024	78.79	**79.41**	**82.20**
	SBERT+DWT_{cD}	512	**78.86**	79.36	82.00
	SBERT+DWT_{cA}	512	78.16	79.30	82.02
	SBERT+DWT_{cAA}	256	76.68	78.93	81.59
	SBERT+DWT_{cDA}	256	78.26	78.95	81.22
$RoBERTa_{Base}$	Baseline	768	80.17	**80.47**	84.04
	RoBERTa+DWT_{cD}	384	**80.34**	80.38	**84.64**
	RoBERTa+DWT_{cA}	384	79.38	80.35	83.68
	RoBERTa+DWT_{cAA}	192	78.76	79.81	82.16
	RoBERTa+DWT_{cDA}	192	79.58	79.95	82.95
$RoBERTa_{Large}$	Baseline	1024	70.34	72.41	77.60
	RoBERTa+DWT_{cD}	512	70.00	72.30	77.51
	RoBERTa+DWT_{cA}	512	**70.34**	72.58	77.58
	RoBERTa+DWT_{cAA}	192	69.68	**72.72**	**77.61**
	RoBERTa+DWT_{cDA}	192	69.52	72.44	76.73

Table 8: Results on the STS13-STS15 benchmark, Spearman's correlation is reported. Baseline represents the original embedding and corresponding performance for SBERT and RoBERTa models. The best overall results are shown in bold. Best results per condition are shown in red.

coefficients outperform the baselines for all tasks by 1.5-3% better performance except for STSB ad SICKR which have comparable results. Level-2 cDA coefficients exceed the performance in STS12,STS15 and STS16 at a dimension reduction by 75% showing (1) that DWT condensed more relevant semantics for these task in subsequent levels of transformation and (2) that the nuance in the features represented in the original embeddings were more relevant for these tasks in the InferSent representation context, having cD outperforms cA coefficients. Still, cA coefficients also beat the baseline.

2- Semantic Similarity Tasks (2013-2015) Subsequently, we present the results for applying DWT on STS tasks 2013-2015 (Agirre et al., 2013, 2014, 2015) using Pre-trained Language Model embeddings, SBERT and RoBERTa, for a detailed study for the performance of DWT on more semantic similarity tasks. Table 8 shows the result of applying DWT embeddings using SBERT and RoBERTa models. As shown, DWT outperforms the baselines for STS13 and is very comparable to STS14 and STS15.

A.2 4-Level DWT Embedding Results

In this section we represent the detailed results for Figure4 for 4-Levels of DWT tasks applied to downstream tasks included in Section6.2 using RoBERTa Large embeddings. Table9 shows the results for Level-1 approximation coefficients, cA, Level-2 approximation coefficients cAA, Level-3 approximation coefficients, cAAA and Level-4 coefficients, cAAAA. The results show the efficacy of DWT approximation coefficients to maintain relevant information despite decreasing the size of the embedding by more than 90%.

Model	**Dim**	**MR**	**CR**	**SST2**	**SST5**	**MPQA**	**SICK-E**	**MRPC**	**SUBJ**	**TREC**
$\text{RoBERTa}_{\text{Large}}$	1024	85.01	91.18	91.38	50.95	90.13	80.68	76.17	92.00	85.80
DWT_{cA}	512	**85.97**	91.21	90.99	51.04	90.62	80.64	77.23	**92.21**	**87.20**
DWT_{cAA}	256	85.49	**91.29**	90.99	**53.85**	**90.72**	**81.33**	77.39	91.80	84.00
DWT_{cAAA}	128	85.22	91.34	**91.54**	52.35	90.22	79.03	77.62	91.02	82.40
$\text{DWT}_{\text{cAAAA}}$	64	84.26	90.62	90.44	50.05	90.34	76.74	**77.51**	88.97	78.40

Table 9: Best Classification accuracy results on various classification tasks for Level-1 approxmation; cA, Level-2 DWT coefficients; cAA, Level-3 DWT coefficients; cAAA and Level-4 DWT coefficients; cAAAA. The Baseline is the original RoBERTa Large model. The best overall results are shown in bold. Best results per condition are shown in red