

SBERT-WK: A Sentence Embedding Method By Dissecting BERT-based Word Models

Bin Wang, *Student Member, IEEE*, and C.-C. Jay Kuo, *Fellow, IEEE*

Abstract—Sentence embedding is an important research topic in natural language processing (NLP) since it can transfer knowledge to downstream tasks. Meanwhile, a contextualized word representation, called BERT, achieves the state-of-the-art performance in quite a few NLP tasks. Yet, it is an open problem to generate a high quality sentence representation from BERT-based word models. It was shown in previous study that different layers of BERT capture different linguistic properties. This allows us to fuse information across layers to find better sentence representation. In this work, we study the layer-wise pattern of the word representation of deep contextualized models. Then, we propose a new sentence embedding method by dissecting BERT-based word models through geometric analysis of the space spanned by the word representation. It is called the SBERT-WK method. No further training is required in SBERT-WK. We evaluate SBERT-WK on semantic textual similarity and downstream supervised tasks. Furthermore, ten sentence-level probing tasks are presented for detailed linguistic analysis. Experiments show that SBERT-WK achieves the state-of-the-art performance. Our codes are publicly available at the Github¹.

Index Terms—Sentence Embedding, Deep Contextualized Models, BERT, Subspace Analysis, Clustering.

I. INTRODUCTION

STATIC word embedding is a popular learning technique that transfers prior knowledge from a large unlabeled corpus [1]–[3]. Most of recent sentence embedding methods are rooted in that static word representations can be embedded with rich syntactic and semantic information. It is desired to extend the word-level embedding to the sentence-level, which contains a longer piece of texts. We have witnessed a breakthrough by replacing the “static” word embedding to the “contextualized” word representation in the last several years, e.g., [4]–[7]. A natural question to ask is how to exploit contextualized word embedding in the context of sentence embedding. Here, we examine the problem of learning the universal representation of sentences. A contextualized word representation, called BERT, achieves the state-of-the-art performance in many natural language processing (NLP) tasks. We aim at developing a sentence embedding solution from BERT-based models in this work.

As reported in [8] and [9], different layers of BERT learn different levels of information and linguistic properties. While

intermediate layers encode the most transferable features, representation from higher layers are more expressive in high-level semantic information. Thus, information fusion across layers has its potential in providing a stronger representation. Furthermore, by conducting experiments on patterns of the isolated word representation across layers in deep models, we observe the following property. Words of richer information in a sentence have higher variation in their representations, while the token representation changes gradually, across layers. This finding helps define “salient” word representations and informative words in computing universal sentence embedding.

Although the BERT-based contextualized word embedding method performs well in NLP tasks [10], it has its own limitations. For example, due to the large model size, it is time consuming to perform sentence pair regression such as clustering and semantic search. The most effective way to solve this problem is through an improved sentence embedding method, which transforms a sentence to a vector that encodes the semantic meaning of the sentence. Currently, a common sentence embedding approach based on BERT-based models is to average the representations obtained from the last layer or using the CLS token for sentence-level prediction. Yet, both are sub-optimal as shown in the experimental section of this paper. To the best of our knowledge, there is only one paper on sentence embedding using pre-trained BERT, called Sentence-BERT or SBERT [11]. It leverages further training with high-quality labeled sentence pairs. Apparently, how to obtain sentence embedding from deep contextualized models is still an open problem.

While word embedding is learned using a loss function defined on word pairs, sentence embedding demands a loss function defined at the sentence-level. Following a path similar to word embedding, unsupervised learning of sentence encoders, e.g., SkipThought [12] and FastSent [13], build self-supervision from a large unlabeled corpus. Yet, InferSent [14] shows that training on high quality labeled data, e.g., the Stanford Natural Language Inference (SNLI) dataset, can consistently outperform unsupervised training objectives. Recently, leveraging training results from multiple tasks has become a new trend in sentence embedding since it provides better generalization performance. USE [15] incorporates both supervised and unsupervised training objectives on the Transformer architecture. The method in [16] is trained in a multi-tasking manner so as to combine inductive biases of diverse training objectives. However, multi-tasking learning for sentence embedding is still under development, and it faces some difficulty in selecting supervised tasks and handling interactions between tasks. Furthermore, supervised training

Bin Wang is with Ming-Hsieh Department of Electrical and Computer Engineering and Signal and Image Processing Institute, University of Southern California, Los Angeles, CA, 90089-2584 USA (e-mail:bwang28c@gmail.com).

C.-C. Jay Kuo is with Ming-Hsieh Department of Electrical and Computer Engineering, Department of Computer Science and Signal and Image Processing Institute, University of Southern California, Los Angeles, CA, 90089-2584 USA (e-mail:ckkuo@sipi.usc.edu).

¹ Link.

objectives demand high quality labeled data which are usually expensive.

Being drastically different from the above-mentioned research, we investigate sentence embedding by studying the geometric structure of deep contextualized models and propose a new method by dissecting BERT-based word models. It is called the SBERT-WK method. As compared with previous sentence embedding models that are trained on sentence-level objectives, deep contextualized models are trained on a large unlabeled corpus with both word- and sentence-level objectives. SBERT-WK inherits the strength of deep contextualized models. It is compatible with most deep contextualized models such as BERT [5] and SBERT [11].

This work has the following three main contributions.

- 1) We study the evolution of isolated word representation patterns across layers in BERT-based models. These patterns are shown to be highly correlated with word's content. It provides useful insights into deep contextualized word models.
- 2) We propose a new sentence embedding method, called SBERT-WK, through geometric analysis of the space learned by deep contextualized models.
- 3) We evaluate the SBERT-WK method against eight downstream tasks and seven semantic textual similarity tasks, and show that it achieves state-of-the-art performance. Furthermore, we use sentence-level probing tasks to shed light on the linguistic properties learned by SBERT-WK.

The rest of the paper is organized as following. Related work is reviewed in Sec. II. The evolution of word representation patterns in deep contextualized models is studied in Sec. III. The proposed SBERT-WK method is presented in Sec. IV. The SBERT-WK method is evaluated with respect to various tasks in Sec. V. Finally, concluding remarks and future work directions are given in Sec. VI.

II. RELATED WORK

A. Contextualized Word Embedding

Traditional word embedding methods provide a static representation for a word in a vocabulary set. Although the static representation is widely adopted in NLP, it has several limitations in modeling the context information. First, it cannot deal with polysemy. Second, it cannot adjust the meaning of a word based on its contexts. To address the shortcomings of static word embedding methods, there is a new trend to go from shallow to deep contextualized representations. For example, ELMo [4], GPT1 [7], GPT2 [17] and BERT [5] are pre-trained deep neural language models, and they can be fine-tuned on specific tasks. These new word embedding methods achieve impressive performance on a wide range of NLP tasks. In particular, the BERT-based models are dominating in leaderboards of language understanding tasks such as SQuAD2.0 [18] and GLUE benchmarks [10].

ELMo is one of the earlier work in applying a pre-trained language model to downstream tasks [4]. It employs two layer bi-directional LSTM and fuses features from all LSTM outputs using task-specific weights. OpenAI GPT [7] incorporates a fine-tuning process when it is applied to downstream tasks.

Tasks-specific parameters are introduced and fine-tuned with all pre-trained parameters. BERT employs the Transformer architecture [19], which is composed by multiple multi-head attention layers. It can be trained more efficiently than LSTM. It is trained on a large unlabeled corpus with several objectives to learn both word- and sentence-level information, where the objectives include masked language modeling as well as the next sentence prediction. A couple of variants have been proposed based on BERT. RoBERTa [20] attempts to improve BERT by providing a better recipe in BERT model training. ALBERT [21] targets at compressing the model size of BERT by introducing two parameter-reduction techniques. At the same time, it achieves better performance. XLNET [6] adopts a generalized auto-regressive pre-training method that has the merits of auto-regressive and auto-encoder language models.

Because of the superior performance of BERT-based models, it is important to have a better understanding of BERT-based models and the transformer architecture. Efforts have been made along this direction recently as reviewed below. Liu *et al.* [9] and Petroni *et al.* [22] used word-level probing tasks to investigate the linguistic properties learned by the contextualized models experimentally. Kovaleva *et al.* [23] and Michel *et al.* [24] attempted to understand the self-attention scheme in BERT-based models. Hao *et al.* [25] provided insights into BERT by visualizing and analyzing the loss landscapes in the fine-tuning process. Ethayarajh [26] explained how the deep contextualized model learns the context representation of words. Despite the above-mentioned efforts, the evolving pattern of a word representation across layers in BERT-based models has not been studied before. In this work, we first examine the pattern evolution of a token representation across layers without taking its context into account. With the context-independent analysis, we observe that the evolving patterns are highly related to word properties. This observation in turn inspires the proposal of a new sentence embedding method – SBERT-WK.

B. Universal Sentence Embedding

By sentence embedding, we aim at extracting a numerical representation for a sentence to encapsulate its meanings. The linguistic features learned by a sentence embedding method can be external information resources for downstream tasks. Sentence embedding methods can be categorized into two categories: non-parameterized and parameterized models. Non-parameterized methods usually rely on high quality pre-trained word embedding methods. The simplest example is to average word embedding results as the representation for a sentence. Following this line of thought, several weighted averaging methods were proposed, including tf-idf, SIF [27], uSIF [28] and GEM [29]. SIF uses the random walk to model the sentence generation process and derives word weights using the maximum likelihood estimation (MLE). uSIF extends SIF by introducing an angular-distance-based random walk model. No hyper-parameter tuning is needed in uSIF. By exploiting geometric analysis of the space spanned by word embeddings, GEM determines word weights with several hand-crafted measurements. Instead of weighted averaging, it

uses the p -mean [30] to concatenate the power means of word embeddings and fuses different word embedding models so as to shorten the performance gap between non-parameterized and parameterized models.

Parameterized models are more complex, and they usually perform better than non-parameterized models. The skip-thought model [12] extends the unsupervised training of word2vec [1] from the word level to the sentence level. It adopts the encoder-decoder architecture to learn the sentence encoder. InferSent [14] employs bi-directional LSTM with supervised training. It trains the model to predict the entailment or contradiction of sentence pairs with the Stanford Natural Language Inference (SNLI) dataset. It achieves better results than methods with unsupervised learning. The USE (Universal Sentence Encoder) method [15] extends the InferSent model by employing the Transformer architecture with unsupervised as well as supervised training objectives. It was observed by later studies [16], [31] that training with multiple objectives in sentence embedding can provide better generalizability.

The SBERT method [11] is the only parameterized sentence embedding model using BERT as the backbone. SBERT shares high similarity with InferSent [14]. It uses the Siamese network on top of the BERT model and fine-tunes it based on high quality sentence inference data (e.g. the SNLI dataset) to learn more sentence-level information. However, unlike supervised tasks, universal sentence embedding methods in general do not have a clear objective function to optimize. Instead of training on more sophisticated multi-tasking objectives, we combine the advantage of both parameterized and non-parameterized methods. SBERT-WK is computed by subspace analysis of the manifold learned by the parameterized BERT-based models.

C. Subspace Learning and Analysis

In signal processing and data science, subspace learning and analysis offer powerful tools for multidimensional data processing. Correlated data of a high dimension can be analyzed using latent variable representation methods such as Principal Component Analysis (PCA), Singular Spectrum Analysis (SSA), Independent Component Analysis (ICA) and Canonical Correlation Analysis (CCA). Subspace analysis has solid mathematical foundation. It is used to explain and understand the internal states of Deep Neural Networks [32], [33], [34].

The goal of word or sentence embedding is to map words or sentences onto a high-dimensional space. Thus, subspace analysis is widely adopted in this field, especially for word embedding. Before learning-based word embedding, the factorization-based word embedding methodology is the mainstream, which is used to analyze the co-occurrence statistics of words. To find word representations, Latent Semantic Analysis (LSA) [35] factorizes the co-occurrence matrix using singular value decomposition. Levy and Goldberg [36] pointed out the connection between the word2vec [1] model and the factorization-based methods. Recently, subspace analysis is adopted for interpretable word embedding because of mathematical transparency. Subspace analysis is also widely used in post-processing and evaluation of word embedding models [37], [38], [39].

Because of the success of subspace analysis in word embedding, it is natural to incorporate subspace analysis in sentence embedding as a sentence is composed by a sequence of words. For example, SCDV [40] determines the sentence/document vector by splitting words into clusters and analyzing them accordingly. GEM [29] models the sentence generation process as a Gram-Schmidt procedure and expands the subspace formed by word vectors gradually. Both DCT [41] and EigenSent [42] map a sentence matrix into the spectral space and model the high-order dynamics of a sentence from a signal processing perspective.

Although subspace analysis has already been applied to sentence embedding, all above-mentioned work was built upon on static word embedding methods. To the best of our knowledge, our work is the first one that exploits subspace analysis to find generic sentence embeddings based on deep contextualized word models. We will show in this work that SBERT-WK can consistently outperform state-of-the-art methods with low computational overhead and good interpretability, which is attributed to high transparency and efficiency of subspace analysis and the power of deep contextualized word embedding.

III. WORD REPRESENTATION EVOLUTION ACROSS LAYERS

Although studies have been done in the understanding of the word representation learned by deep contextualized models, none of them examine how a word representation evolves across layers. To observe such an evolving pattern, we design experiments in this section by considering the following four BERT-based models.

- BERT [5]. It employs the bi-directional training of the transformer architecture and applies it to language modeling. Unsupervised objectives, including the masked language model and the next sentence prediction, are incorporated.
- SBERT [11]. It integrates the Siamese network with a pre-trained BERT model. The supervised training objective is added to learn high quality sentence embedding.
- RoBERTa [20]. It adapts the training process of BERT to more general environments such as longer sequences, bigger batches, more data and mask selection schemes, etc. The next sentence prediction objective is removed.
- XLNET [6]. It adopts the Transformer-XL architecture, which is trained with the Auto-Regressive (AR) objective.

The above four BERT-based models have two variants; namely, the 12-layer base model and the 24-layer large model. We choose their base models in the experiments, which are pre-trained on their respective language modeling tasks.

To quantify the evolution of word representations across layers of deep contextualized models, we measure the pair-wise cosine similarity between 1- and N -hop neighbors. By the 1-hop neighbor, we refer to the representation in the preceding or the succeeding layer of the current layer. Generally, word w has $(N + 1)$ representations of dimension d for a N -layer transformer network. The whole representation set for w can be expressed as

$$v_w^0, v_w^1, \dots, v_w^N, \quad (1)$$

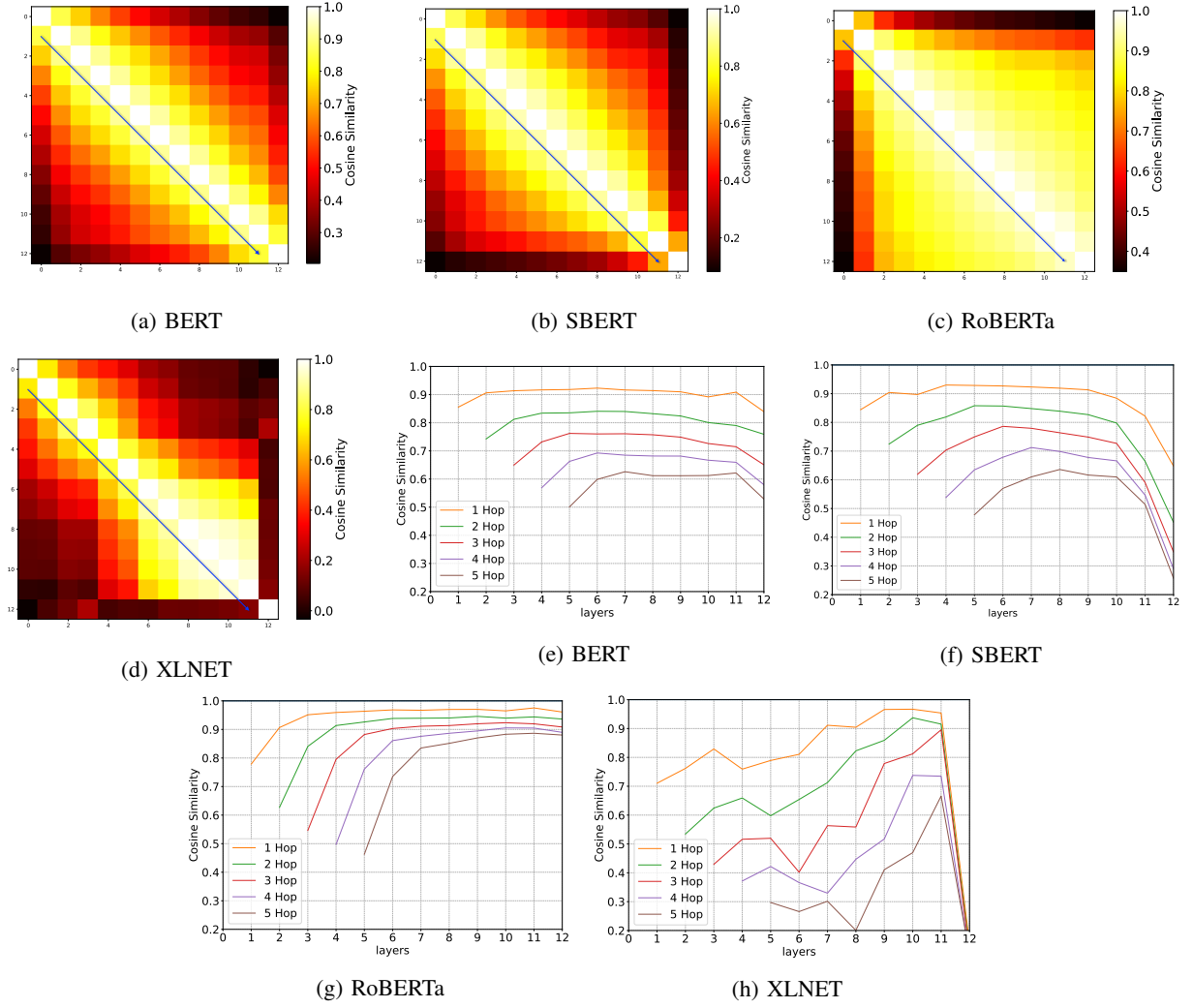


Fig. 1: Evolving word representation patterns across layers measured by cosine similarity, where (a-d) show the similarity across layers and (e-h) show the similarity over different hops. Four contextualized word representation models (BERT, SBERT, RoBERTa and XLNET) are tested.

where $v_w^i \in \mathbb{R}^d$ denotes the representation of word w at the i -th layer. The pair-wise cosine similarity between representations of the i -th and the j -th layers can be computed as

$$\text{CosSim}(i, j) = \frac{\langle v_w^i, v_w^j \rangle}{|v_w^i| |v_w^j|}. \quad (2)$$

To obtain statistical results, we extract word representations from all sentences in the popular STSbenchmark dataset [43]. The dataset contains 8628 sentence pairs from three categories: captions, news and forum. The similarity map is non-contextualized. We average the similarity map for all words to present the pattern for contextualized word embedding models.

Figs. 1 (a)-(d) show the similarity matrix across layers for four different models. Figs. 1 (e)-(h) show the patterns along the offset diagonal. In general, we see that the representations from nearby layers share a large similarity value except for that in the last layer. Furthermore, we observe that, except for the main diagonal, offset diagonals do not have an uniform pattern as indicated by the blue arrow in the associated figure. For BERT, SBERT and RoBERTa, the patterns at intermediate

layers are flatter as shown in Figs. 1 (e)-(g). The representations between consecutive layers have a cosine similarity value that larger than 0.9. The rapid change mainly comes from the beginning and the last several layers of the network. This explains why the middle layers are more transferable to other tasks as observed in [9]. Since the representation in middle layers are more stable, more generalizable linguistic properties are learned there. As compared with BERT, SBERT and RoBERTa, XLNET has a very different evolving pattern of word representations. Its cosine similarity curve as shown in Fig. 1 (h) is not concave. This can be explained by the fact that XLNET deviates from BERT significantly from architecture selection to training objectives. It also sheds light on why SBERT [11], which has XLNET as the backbone for sentence embedding generation, has sentence embedding results worse than BERT, given that XLNET is more powerful in other NLP tasks.

We see from Figs. 1 (e)-(g) that the word representation evolving patterns in the lower and the middle layers of BERT,

TABLE I: Word groups based on the variance level. Less significant words in a sentence are underlined.

Variance	Low	Middle	High
SBERT	<u>can</u> , <u>end</u> , <u>do</u> , <u>would</u> , time, <u>all</u> , say, says, <u>how</u> , <u>before</u> , more, east, <u>be</u> , <u>have</u> , <u>so</u> , <u>could</u> , <u>that</u> , <u>than</u> , <u>been</u> , south, united, <u>what</u> , peace, <u>to</u> , states, against, <u>since</u> , <u>first</u> , <u>last</u>	percent, security, <u>mr</u> , <u>into</u> , military, <u>she</u> , arms, <u>they</u> , nuclear, head, billion, <u>on</u> , <u>another</u> , <u>around</u> , <u>their</u> , million, killed, mandela, arrested, wearing, three, <u>his</u> , <u>her</u> , city, <u>through</u> , cutting, green, oil	eating, walking, small, room, person, children, grass, baby, cat, bike, field, runs, potato, horse, snow, ball, dogs, dancing men, dog, running, women, boy, jumping, plane, train, man, camera, woman, guitar
BERT	<u>have</u> , <u>his</u> , <u>their</u> , <u>last</u> , runs, <u>would</u> <u>been</u> , running, <u>all</u> , <u>than</u> , she, <u>that</u> <u>to</u> , cat, boy, <u>be</u> , <u>first</u> , woman, <u>how</u> cutting, <u>since</u> , dogs, dog, say, <u>could</u> , <u>more</u> , man, small, eating	jumping, <u>on</u> , against, <u>into</u> , man, baby <u>around</u> , walking, person, green, <u>her</u> , <u>end</u> , <u>through</u> , <u>another</u> , three, <u>so</u> , wearing, mandela, south, <u>do</u> , potato, <u>they</u> , <u>what</u> , women, says, <u>can</u> , arrested	military, nuclear, killed, dancing, percent peace, plane, united, <u>mr</u> , bike, guitar, oil, train, children, arms, east, camera grass, ball, field, room, horse, <u>before</u> , billion city, security, million, snow, states, time

SBERT and RoBERTa are quite similar. Their differences mainly lie in the last several layers. SBERT has the largest drop while RoBERTa has the minimum change in cosine similarity measures in the last several layers. SBERT has the highest emphasis on the sentence-pair objective since it uses the Siamese network for sentence pair prediction. BERT puts some focus on the sentence-level objective via next-sentence prediction. In contrast, RoBERTa removes the next sentence prediction completely in training.

We argue that faster changes in the last several layers are related to the training with the sentence-level objective, where the distinct sentence level information is reflected. Generally speaking, if more information is introduced by a word, we should pay special attention to its representation. To quantify such a property, we propose two metrics (namely, alignment and novelty) in Sec. IV-A.

We have so far studied the evolving pattern of word representations across layers. We may ask whether such a pattern is word dependent. This question can be answered below. As shown in Fig. 1, the offset diagonal patterns are pretty similar with each other in the mean. Without loss of generality, we conduct experiments on the offset-1 diagonal that contains 12 values as indicated by the arrow in Fig. 1. We compute the variances of these 12 values to find the variability of the 1-hop cosine similarity values with respect to different words. The variance is computed for each word in BERT and SBERT². We only report words that appears more than 50 times to avoid randomness in Table I. The same set of words were reported for BERT and SBERT models. The words are split into three categorizes based on their variance values. The insignificant words in a sentence are underlined. We can clearly see from the table that words in the low variance group are in general less informative. In contrast, words in the high variance group are mostly nouns and verbs, which usually carry richer content. We conclude that more informative words in deep contextualized models vary more while insignificant words vary less. This finding motivates us to design a module that can distinguish important words in a sentence in Sec. IV-B.

²Since RoBERTa and XLNET use a special tokenizer, which cannot be linked to real word pieces, we do not test on RoBERTa and XLNET here.

IV. PROPOSED SBERT-WK METHOD

We propose a new sentence embedding method called SBERT-WK in this section. The block diagram of the SBERT-WK method is shown in Fig. 2. It consists of the following two steps:

- 1) Determine a unified word representation for each word in a sentence by integrating its representations across layers by examining its alignment and novelty properties.
- 2) Conduct a weighted average of unified word representations based on the word importance measure to yield the ultimate sentence embedding vector.

They are elaborated in the following two subsections, respectively.

A. Unified Word Representation Determination

As discussed in Sec. III, the word representation evolves across layers. We use v_w^i to denote the representation of word w at the i th layer. To determine the unified word representation, \hat{v}_w , of word w in Step 1, we assign weight α_i to its i -th layer representation, v_w^i , and take an average:

$$\hat{v}_w = \sum_{i=0}^N \alpha(v_w^i) v_w^i, \quad (3)$$

where weight α can be derived based on the inverse alignment and the novelty two properties.

1) *Inverse Alignment Measure*: We define the context matrix of v_w^i as

$$\mathbf{C} = [v_w^{i-m}, \dots, v_w^{i-1}, v_w^i, v_w^{i+1}, \dots, v_w^{i+m}] \in \mathbb{R}^{d \times 2m}, \quad (4)$$

where d is the word embedding dimension and m is the context window size. We can compute the pair-wise cosine similarity between v_w^i and all elements in the context window $C(v_w^i)$ and use their average to measure how v_w^i aligns with the word vectors inside its context context. Then, the alignment similarity score of v_w^i can be defined as

$$\beta_a(v_w^i) = \frac{1}{2m} \sum_{j=i-m, j \neq i}^{i+m} \frac{\langle v_w^i, v_w^j \rangle}{|v_w^i| |v_w^j|}. \quad (5)$$

If a word representation at a layer aligns well with its context word vectors, it does not provide much additional information. Since it is less informative, we can give it a smaller weight.

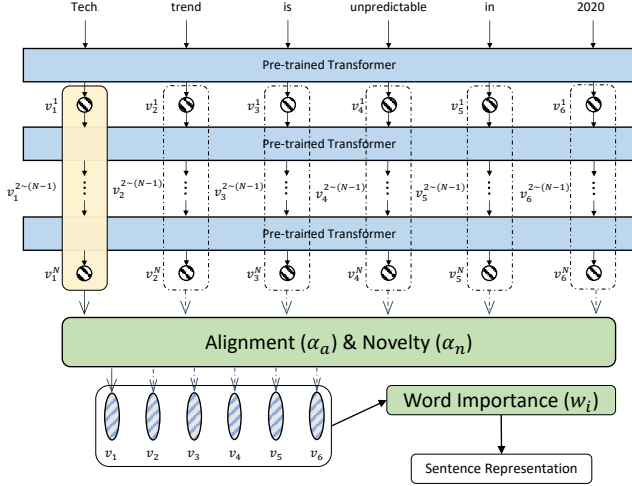


Fig. 2: Illustration for the proposed SBERT-WK model.

Thus, we use the inverse of the alignment similarity score as the weight for word w at the i -th layer. Mathematically, we have

$$\alpha_a(v_w^i) = \frac{K_a}{\beta_a(v_w^i)}, \quad (6)$$

where K_a is a normalization constant independent of i and it is chosen to normalize the sum of weights:

$$\sum_{i=1}^N \alpha_a(v_w^i) = 1.$$

We call $\alpha_a(v_w^i)$ the inverse alignment weight.

2) *Novelty Measure*: Another way to measure the new information of word representation v_w^i is to study the new information brought by it with respect to the subspace spanned words in its context window. Clearly, words in the context matrix \mathbf{C} form a subspace. We can decompose v_w^i into two components: one contained by the subspace and the other orthogonal to the subspace. We view the orthogonal one as its novel component and use its magnitude as the novelty score. By singular value decomposition (SVD), we can factorize matrix \mathbf{M} of dimension $m \times n$ into the form $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$, where \mathbf{U} is an $m \times n$ matrix with orthogonal columns, $\mathbf{\Sigma}$ is an $n \times n$ diagonal matrix with non-negative numbers on the diagonal and \mathbf{V} is $n \times n$ orthogonal matrix. In our current context, we decompose the context matrix \mathbf{C} in Eq. (4) to $\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$ to find the orthogonal basis for the context words. The orthogonal column basis for \mathbf{C} is represented by matrix \mathbf{U} . Thus, the orthogonal component of v_w^i with respect to \mathbf{C} can be computed as

$$q_w^i = v_w^i - \mathbf{U}\mathbf{U}^T v_w^i. \quad (7)$$

The novelty score of v_w^i is computed by

$$\alpha_n(v_w^i) = \frac{K_n \|q_w^i\|_2}{\|v_w^i\|_2}, \quad (8)$$

where K_n is a normalization constant independent of i and it is chosen to normalize the sum of weights:

$$\sum_{i=1}^N \alpha_n(v_w^i) = 1.$$

We call $\alpha_n(v_w^i)$ the novelty weight.

3) *Unified Word Representation*: We examine two ways to measure the new information brought by word representation v_w^i at the i -th layer. We may consider a weighted average of the two in form of

$$\alpha_c(v_w^i, \omega) = \omega \alpha_a(v_w^i) + (1 - \omega) \alpha_n(v_w^i), \quad (9)$$

where $0 \leq \omega \leq 1$ and $\alpha_c(v_w^i, \omega)$ is called the combined weight. We compare the performance of three cases (namely, novelty weight $\omega = 0$, inverse alignment weight $\omega = 1$ and combined weight $\omega = 0.5$) in the experiments. A unified word representation is computed as a weighted sum of its representations in different layers:

$$\hat{v}_w = \sum_{i=0}^N \alpha_c(v_w^i) v_w^i. \quad (10)$$

We can view v_w as the new contextualized word representation for word w .

B. Word Importance

As discussed in Sec. III, the variances of the pair-wise cosine-similarity matrix can be used to categorize words into different groups. Words of richer information usually have a larger variance. By following the line of thought, we can use the same variance to determine the importance of a word and merge multiple words in a sentence to determine the sentence embedding vector. This is summarized below.

For the j -th word in a sentence denoted by $w(j)$, we first compute its cosine similarity matrix using its word representations from all layers as shown in Eq. (2). Next, we extract the offset-1 diagonal of the cosine similarity matrix, compute the variance of the offset-1 diagonal values and use σ_j^2 to denote the variance of the j th word. Then, the final sentence embedding (v_s) can be expressed as

$$v_s = \sum_j \omega_j \hat{v}_{w(j)}, \quad (11)$$

where $\hat{v}_{w(j)}$ is the new contextualized word representation for word $w(j)$ as defined in Eq. (10) and

$$\omega_j = \frac{|\sigma_j^2|}{\sum_k |\sigma_k^2|}. \quad (12)$$

Note that the weight for each word is the l_1 -normalized variance as shown in Eq. 12. To sum up, in our sentence embedding scheme, words that evolve faster across layers with get higher weights since they have larger variances.

C. Computational Complexity

The main computational burden of SBERT-WK comes from the SVD decomposition, which allows more fine-grained analysis in novelty measure. The context window matrix \mathbf{C} is decomposed into the product of three matrices $\mathbf{C} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$. The orthogonal basis is given by matrix \mathbf{U} . The context window matrix is of size $d \times 2m$, where d is the word embedding size and $2m$ is the whole window size. In our case, d is much larger than m so that the computational complexity for SVD is $O(8dm^2)$, where several terms are ignored.

Instead of performing SVD decomposition, we use the QR factorization in our experiments as an alternative because of its computational efficiency. With QR factorization, we first concatenate the center word vector representation v_w^i to the context window matrix \mathbf{C} to form a new matrix

$$\tilde{\mathbf{C}} = [v_w^{i-m}, \dots, v_w^{i-1}, v_w^{i+1}, \dots, v_w^{i+m}, v_w^i] \in \mathbb{R}^{d \times (2m+1)} \quad (13)$$

has $2m + 1$ word representations. We perform the QR factorization on $\tilde{\mathbf{C}}$, and obtain $\tilde{\mathbf{C}} = \mathbf{Q}\mathbf{R}$, where non-zero columns of matrix $\mathbf{Q} \in \mathbb{R}^{d \times (2m+1)}$ are orthonormal basis and $\mathbf{R} \in \mathbb{R}^{(2m+1) \times (2m+1)}$ is an upper triangular matrix that contains the weights for word representations under the basis of \mathbf{Q} . We denote the i th column of \mathbf{Q} and \mathbf{R} as \mathbf{q}^i and \mathbf{r}^i , respectively. With QR factorization, \mathbf{r}^{2m+1} is the representation of v_w^i under the orthogonal basis formed by matrix \mathbf{Q} . The new direction introduced to the context by v_w^i is represented as q^{2m+1} . Then, the last component of \mathbf{r}^{2m+1} is the weight for the new direction, which is denoted by r_{-1}^{2m+1} . Then, the novelty weight can be derived as:

$$\alpha_n(v_w^i) = \frac{K_n r_{-1}^{2m+1}}{|r^{2m+1}|}, \quad (14)$$

where K_n is the normalization constant. The inverse alignment weight can also be computed under the new basis \mathbf{Q} .

The complexity of the QR factorization is $O(d(2m+1)^2)$, which is two times faster than the SVD decomposition. In practice, we see little performance difference between these two methods. The experimental runtime is compared in Sec. V-E.

V. EXPERIMENTS

Since our goal is to obtain a general purpose sentence embedding method, we evaluate SBERT-WK on three kinds of evaluation tasks.

- Semantic textual similarity tasks.
They predict the similarity between two given sentences. They can be used to indicate the embedding ability of a method in terms of clustering and information retrieval via semantic search.
- Supervised downstream tasks.
They measure embedding's transfer capability to downstream tasks including entailment and sentiment classification.
- Probing tasks.
They are proposed in recent years to measure the linguistic features of an embedding model and provide fine-grained analysis.

These three kinds of evaluation tasks can provide a comprehensive test on our proposed model. The popular SentEval toolkit [44] is used in all experiments. The proposed SBERT-WK method can be built upon several state-of-the-art pre-trained language models such as BERT, SBERT, RoBERTa and XLNET. Here, we evaluate it on top of two models: SBERT³ and XLNET⁴. The latter is the XLNET pre-trained model obtained from the Transformer Repo. We adopt their base models that contain 12 self-attention layers.

For performance benchmarking, we compare SBERT-WK with the following 10 different methods, including parameterized and non-parameterized models.

- 1) Average of GloVe word embeddings;
- 2) Average of FastText word embedding;
- 3) Average the last layer token representations of BERT;
- 4) Use [CLS] embedding from BERT, where [CLS] is used for next sentence prediction in BERT;
- 5) SIF model [27], which is a non-parameterized model that provides a strong baseline in textual similarity tasks;
- 6) p -mean model [30] that incorporate multiple word embedding models;
- 7) Skip-Thought [12];
- 8) InferSent [27] with both GloVe and FastText versions;
- 9) Universal Sentence Encoder [15], which is a strong parameterized sentence embedding using multiple objectives and transformer architecture;
- 10) Sentence-BERT, which is a state-of-the-art sentence embedding model by training the Siamese network over BERT.

A. Semantic Textual Similarity

To evaluate semantic textual similarity, we use 2012-2016 STS datasets [45]–[49]. They contain sentence pairs and labels between 0 and 5, which indicate their semantic relatedness. Some methods learn a complex regression model that maps sentence pairs to their similarity score. Here, we use the cosine similarity between sentence pairs as the similarity score and report the Pearson correlation coefficient. More details of these datasets can be found in Table II.

Semantic relatedness is a special kind of similarity task, and we use the SICK-R [50] and the STS Benchmark dataset [43] in our experiments. Being different from STS12-STSB16, the semantic relatedness datasets are under the supervised setting where we learn to predict the probability distribution of relatedness scores. The STS Benchmark dataset is a popular dataset to evaluate supervised STS systems. It contains 8,628 sentences from three categories (captions, news and forums) and they are divided into train (5,749), dev (1,500) and test (1,379). The Pearson correlation coefficient is reported.

In our experiments, we do not include the representation from the first three layers since their representations are less contextualized as reported in [26]. Some superficial information is captured by those representations and they play a subsidiary role in most tasks [8]. We set the context window size to $m = 2$ in all evaluation tasks.

³SBERT

⁴Huggingface Transformer 2.2.1

TABLE II: Examples in STS12-STS16, STS-B and SICK datasets.

Dataset	Task	Sent A	Sent B	Label
STS12-STS16	STS	"Don't cook for him. He's a grown up."	"Don't worry about him. He's a grown up."	4.2
STS-B	STS	"Swimmers are racing in a lake."	"Women swimmers are diving in front of the starting platform."	1.6
SICK-R	STS	"Two people in snowsuits are lying in the snow and making snow angel."	"Two angels are making snow on the lying children"	2.5

TABLE III: Experimental results on various textual similarity tasks in terms of the Pearson correlation coefficients (%), where the best results are shown in bold face.

Model	Dim	STS12	STS13	STS14	STS15	STS16	STSB	SICK-R	Avg.
<i>Non-Parameterized models</i>									
Avg. GloVe embeddings	300	52.3	50.5	55.2	56.7	54.9	65.8	80.0	59.34
Ave. FastText embedding	300	58.0	58.0	65.0	68.0	64.0	70.0	82.0	66.43
SIF	300	56.2	56.6	68.5	71.7	-	72.0	86.0	68.50
<i>p</i> -mean	3600	54.0	52.0	63.0	66.0	67.0	72.0	86.0	65.71
<i>Parameterized models</i>									
Skip-Thought	4800	41.0	29.8	40.0	46.0	52.0	75.0	86.0	52.83
InferSent-GloVe	4096	59.3	58.8	69.6	71.3	71.5	75.7	88.4	70.66
InferSent-FastText	4096	62.7	54.8	68.4	73.6	71.8	78.5	88.8	71.23
Universal Sentence Encoder	512	61.0	64.0	71.0	74.0	74.0	78.0	86.0	72.57
BERT [CLS]	768	27.5	22.5	25.6	32.1	42.7	52.1	70.0	38.93
Avg. BERT embedding	768	46.9	52.8	57.2	63.5	64.5	65.2	80.5	61.51
Sentence-BERT	768	64.6	67.5	73.2	74.3	70.1	74.1	84.2	72.57
Proposed SBERT-WK	768	70.2	68.1	75.5	76.9	74.5	80.0	87.4	76.09

The results are given in Table III. We see that the use of BERT outputs directly generates rather poor performance. For example, the CLS token representation gives an average correlation score of 38.93% only. Averaging BERT outputs provides an average correlation score of 61.51%. This is used as the default setting of generating sentence embedding from BERT in the bert-as-service toolkit⁵. They are both worse than non-parameterized models such as averaging FastText word embedding, which is a static word embedding scheme. Their poor performance could be attributed to that the model is not trained using a similar objective function. The masked language model and next sentence prediction objectives are not suitable for a linear integration of representations. The study in [51] explains how linearity is exploited in static word embeddings (e.g., word2vec) and it sheds light on contextualized word representations as well. Among the above two methods, we recommend averaging BERT outputs because it captures more inherent structure of the sentence while the CLS token representation is more suitable for some downstream classification tasks as shown in Table V.

We see from Table III that InferSent, USE and SBERT provide the state-of-the-art performance on textual similarity tasks. Especially, InferSent and SBERT have a mechanism to incorporate the joint representation of two sentences such as the point-wise difference or the cosine similarity. Then, the training process learns the relationship between sentence representations in a linear manner and compute the correlation using the cosine similarity, which is a perfect fit. Since the original BERT model is not trained in this manner, the use of the BERT representation directly would give rather poor performance. Similarly, the XLNET model does not provide satisfactory results in STS tasks.

As compared with other methods, SBERT-WK improves the performance on textual similarity tasks by a significant margin. It is worthwhile to emphasize that we use only 768-dimension vectors for sentence embedding while InferSent uses 4096-dimension vectors. As explained in [14], [30], [52], the increase in the embedding dimension leads to increased performance for almost all models. This may explain SBERT-WK is slightly inferior to InferSent on the SICK-R dataset. For all other tasks, SBERT-WK achieves substantial better performance even with a smaller embedding size.

B. Supervised Downstream Tasks

For supervised tasks, we compare SBERT-WK with other sentence embedding methods in the following eight downstream tasks.

- MR: Binary sentiment prediction on movie reviews [53].
- CR: Binary sentiment prediction on customer product reviews [54].
- SUBJ: Binary subjectivity prediction on movie reviews and plot summaries [55].
- MPQA: Phrase-level opinion polarity classification [56].
- SST2: Stanford Sentiment Treebank with binary labels [57].
- TREC: Question type classification with 6 classes [58].
- MRPC: Microsoft Research Paraphrase Corpus for paraphrase prediction [59].
- SICK-E: Natural language inference dataset [50].

More details on these datasets are given in Table IV.

The design of our sentence embedding model targets at the transfer capability to downstream tasks. Typically, one can tailor a pre-trained language model to downstream tasks through tasks-specific fine-tuning. It was shown in previous work [27], [29] that subspace analysis methods are more

⁵<https://github.com/hanxiao/bert-as-service>

powerful in semantic similarity tasks. However, we would like to show that sentence embedding can provide an efficient way for downstream tasks as well. In particular, we demonstrate that SBERT-WK does not hurt the performance of pre-trained language models. Actually, it can even perform better than the original model in downstream tasks under the SBERT and XLNET backbone settings.

For SBERT-WK, we use the same setting as the one in semantic similarity tasks. For downstream tasks, we adopt a multi-layer-perception (MLP) model that contains one hidden layer of 50 neurons. The batch size is set to 64 and the Adam optimizer is adopted in the training. All experiments are trained with 4 epochs. For MR, CR, SUBJ, MPQA and MRPC, we use the nested 10-fold cross validation. For SST, we use the standard validation. For TREC and SICK-E, we use the cross validation.

The experimental results on eight supervised downstream tasks are given in Table V. Although it is desired to fine-tune deep models for downstream tasks, we see that SBERT-WK still achieves good performance without any fine-tuning. As compared with the other 12 benchmarking methods, SBERT-WK has the best performance in 5 out of the 8 tasks. For the remaining 3 tasks, it still ranks among the top three. SBERT-WK with SBERT as the backbone achieves the best averaged performance (87.90%). The average accuracy improvements over XLNET and SBERT alone are 4.66% and 1.91%, respectively. For TREC, SBERT-WK is inferior to the two best models, USE and BERT[CLS], by 0.6%. For comparison, the baseline SBERT is much worse than USE, and SBERT-WK outperforms SBERT by 6.8%. USE is particularly suitable for TREC since it is pre-trained on question answering data, which is highly related to the question type classification task. In contrast, SBERT-WK is not trained or fine-tuned on similar tasks. For SICK-E, SBERT-WK is inferior to two InferSent-based methods by 1.2%, which could be attributed to the much larger dimension of InferSent.

We observe that averaging BERT outputs and CLS vectors give pretty similar performance. Although CLS provides poor performance for semantic similarity tasks, CLS is good at classification tasks. This is because that the classification representation is used in its model training. Furthermore, the use of MLP as the inference tool would allow certain dimensions to have higher importance in the decision process. The cosine similarity adopted in semantic similarity tasks treats all dimension equally. As a result, averaging BERT outputs and CLS token representation are not suitable for semantic similarity tasks. If we plan to apply the CLS representation and/or averaging BERT outputs to semantic textual similarity, clustering and retrieval tasks, we need to learn an additional transformation function with external resources.

C. Probing Tasks

It is difficult to infer what kind of information is present in sentence representation based on downstream tasks. Probing tasks focus more on language properties and, therefore, help us understand sentence embedding models. We compare SBERT-WK on 10 probing tasks so as to cover a wide range of aspects

from superficial properties to deep semantic meanings. They are divided into three types [60]: 1) surface information, 2) syntactic information and 3) semantic information.

- Surface Information
 - SentLen: Predict the length range of the input sentence with 6 classes.
 - WC: Predict which word is in the sentence given 1000 candidates.
- Syntactic Information
 - TreeDepth: Predict depth of the parsing tree.
 - TopConst: Predict top-constituents of parsing tree within 20 classes.
 - BShift: Predict whether a bigram has been shifted or not.
- Semantic Information
 - Tense: Classify the main clause tense with past or present.
 - SubjNum: Classify the subject number with singular or plural.
 - ObjNum: Classify the object number with singular or plural.
 - SOMO: Predict whether the noun/verb has been replaced by another one with the same part-of-speech character.
 - CoordInv: Sentences are made of two coordinate clauses. Predict whether it is inverted or not.

We use the same experimental setting as that used for supervised tasks. The MLP model has one hidden layer of 50 neurons. The batch size is set to 64 while Adam is used as the optimizer. All tasks are trained in 4 epochs. The standard validation is employed. Being Different from the work in [61] that uses logistic regression for the WC task in the category of surface information, we use the same MLP model to provide simple yet fair comparison.

The performance is shown in Table VI. We see that SBERT-WK yields better results than SBERT in all tasks. Furthermore, SBERT-WK offers the best performance in four of the ten tasks. As discussed in [60], there is a tradeoff in shallow and deep linguistic properties in a sentence. That is, lower layer representations carry more surface information while deep layer representations represent more semantic meanings [8]. By merging information from various layers, SBERT-WK can take care of these different aspects.

The correlation between probing tasks and downstream tasks were studied in [60]. They found that most downstream tasks only correlates with a subset of the probing tasks. WC is positively correlated with all downstream tasks. This indicates that the word content (WC) in a sentence is the most important factor among all linguistic properties. However, in our finding, although p -means provides the best WC performance, it is not the best one in downstream tasks. Based on the above discussion, we conclude that “good performance in WC alone does not guarantee satisfactory sentence embedding and we should pay attention to the high level semantic meaning as well”. Otherwise, averaging one-hot word embedding would give perfect performance, which is however not true.

TABLE IV: Datasets used in supervised downstream tasks.

Dataset	# Samples	Task	Class	Example	Label
MR	11k	movie review	2	"A fascinating and fun film."	pos
CR	4k	product review	2	"No way to contact their customer service"	neg
SUBJ	10k	subjectivity/objectivity	2	"She's an artist, but hasn't picked up a brush in a year."	objective
MPQA	11k	opinion polarity	2	"strong sense of justice"	pos
SST2	70k	sentiment	2	"At first, the sight of a blind man directing a film is hilarious, but as the film goes on, the joke wears thin."	neg
TREC	6k	question-type	6	"What is the average speed of the horses at the Kentucky Derby?"	NUM:speed
MRPC	5.7k	paraphrase detection	2	"The author is one of several defense experts expected to testify." "Spitz is expected to testify later for the defense."	Paraphrase
SICK-E	10k	entailment	3	"There is no woman using an eye pencil and applying eye liner to her eyelid." "A woman is applying cosmetics to her eyelid."	Contradiction

TABLE V: Experimental results on eight supervised downstream tasks, where the best results are shown in bold face.

Model	Dim	MR	CR	SUBJ	MPQA	SST2	TREC	MRPC	SICK-E	Avg.
<i>Non-Parameterized models</i>										
Avg. GloVe embeddings	300	77.9	79.0	91.4	87.8	81.4	83.4	73.2	79.2	81.66
Ave. FastText embedding	300	78.3	80.5	92.4	87.9	84.1	84.6	74.6	79.5	82.74
SIF	300	77.3	78.6	90.5	87.0	82.2	78.0	-	84.6	82.60
<i>p</i> -mean	3600	78.3	80.8	92.6	73.2	84.1	88.4	73.2	83.5	81.76
<i>Parameterized models</i>										
Skip-Thought	4800	76.6	81.0	93.3	87.1	81.8	91.0	73.2	84.3	83.54
InferSent-GloVe	4096	81.8	86.6	92.5	90.0	84.2	89.4	75.0	86.7	85.78
InferSent-FastText	4096	79.0	84.1	92.9	89.0	84.1	92.4	76.4	86.7	85.58
Universal Sentence Encoder	512	80.2	86.0	93.7	87.0	86.1	93.8	72.3	83.3	85.30
BERT [CLS] vector	768	82.3	86.9	95.4	88.3	86.9	93.8	72.1	73.8	84.94
Avg. BERT embedding	768	81.7	86.8	95.3	87.8	86.7	91.6	72.5	78.2	85.08
Avg. XLNET embedding	768	81.6	85.7	93.7	87.1	85.1	88.6	66.5	56.7	80.63
Sentence-BERT	768	82.4	88.9	93.9	90.1	88.4	86.4	75.5	82.3	85.99
SBERT-WK (XLNET)	768	83.6	87.4	94.9	89.1	87.1	91.0	74.2	75.0	85.29
SBERT-WK (SBERT)	768	83.0	89.1	95.2	90.6	89.2	93.2	77.4	85.5	87.90

TABLE VI: Experimental results on 10 probing tasks, where the best results are shown in bold face.

		Surface		Syntactic			Semantic				
Model	Dim	SentLen	WC	TreeDepth	TopConst	BShift	Tense	SubjNum	ObjNum	SOMO	CoordInv
Non-Parameterized models											
Avg. GloVe embeddings	300	71.77	80.61	36.55	66.09	49.90	85.33	79.26	77.66	53.15	54.15
Ave. FastText embedding	300	64.13	82.10	36.38	66.33	49.67	87.18	80.79	80.26	49.97	52.25
<i>p</i> -mean	3600	86.42	98.85	38.20	61.66	50.09	88.18	81.73	83.27	53.27	50.45
Parameterized models											
Skip-Thought	4800	86.03	79.64	41.22	82.77	70.19	90.05	86.06	83.55	54.74	71.89
InferSent-GloVe	4096	84.25	89.74	45.13	78.14	62.74	88.02	86.13	82.31	60.23	70.34
InferSent-FastText	4096	83.36	89.50	40.78	80.93	61.81	88.52	86.16	83.76	53.75	69.47
Universal Sentence Encoder	512	79.84	54.19	30.49	68.73	60.52	86.15	77.78	74.60	58.48	58.19
BERT [CLS] vector	768	68.05	50.15	34.65	75.93	86.41	88.81	83.36	78.56	64.87	74.32
Avg. BERT embedding	768	84.08	61.11	40.08	73.73	88.80	88.74	85.82	82.53	66.76	72.59
Avg. XLNET embedding	768	67.93	42.60	35.84	73.98	72.54	87.29	85.15	78.52	59.15	67.61
Sentence-BERT	768	75.55	58.91	35.56	61.49	77.93	87.32	79.76	78.40	62.85	65.34
SBERT-WK (XLNET)	768	79.91	60.39	43.34	80.70	79.02	88.68	88.16	84.01	61.15	71.71
SBERT-WK (SBERT)	768	92.40	77.50	45.40	79.20	87.87	88.88	86.45	84.53	66.01	71.87

The TREC dataset is shown to be highly correlated with a wide range of probing tasks in [60]. SBERT-WK is better than SBERT in all probing tasks and we expect it to yield excellent performance for the TREC dataset. This is verified in Table V. We see that SBERT-WK works well for the TREC dataset with substantial improvement over the baseline SBERT model.

SBERT is trained using the Siamese Network on top of the BERT model. It is interesting to point out that SBERT underperforms BERT in probing tasks consistently. This could be attributed to that SBERT pays more attention to the sentence-level information in its training objective. It focuses more on sentence pair similarities. In contrast, the mask language objective in BERT focuses more on word- or phrase-level

and the next sentence prediction objective captures the inter-sentence information. Probing tasks are tested on the word-level information or the inner structure of a sentence. They are not well captured by the SBERT sentence embedding. Yet, SBERT-WK can enhance SBERT significantly through detailed analysis of each word representation. As a result, SBERT-WK can obtain similar or even better results than BERT in probing tasks.

D. Ablation and Sensitivity Study

To verify the effectiveness of each module in the proposed SBERT-WK model, we conduct the ablation study by adding one module at a time. Also, the effect of two hyper parameters

(the context window size and the starting layer selection) is evaluated. The averaged results for textual semantic similarity datasets, including STS12-STS16 and STSB, are presented.

1) *Ablation study of each module's contribution:* We present the ablation study results in Table VII. It shows that all three components (Alignment, Novelty, Token Importance) improve the performance of the plain SBERT model. Adding the Alignment weight and the Novelty weight alone provides performance improvement of 1.86% and 2.49%, respectively. The Token Importance module can be applied to the word representation of the last layer or the word representation obtained by averaging all layer outputs. The corresponding improvements are 0.55% and 2.2%, respectively. Clearly, all three modules contribute to the performance of SBERT-WK. The ultimate performance gain can reach 3.56%.

TABLE VII: Comparison of different configurations to demonstrate the effectiveness of each module of the proposed SBERT-WK method. The averaged Pearson correlation coefficients (%) for STS12-STS16 and STSB datasets are reported.

Model	Avg. STS results
SBERT baseline	70.65
SBERT + Alignment ($w = 0$)	72.51
SBERT + Novelty ($w = 1$)	73.14
SBERT + Token Importance (last layer)	71.20
SBERT + Token Importance (all layers)	72.85
SBERT-WK ($w = 0.5$)	74.21

2) *Sensitivity to window size and layer selection:* We test the sensitivity of SBERT-WK to two hyper-parameters on STS, SICK-E and SST2 datasets. The results are shown in Fig. 3. The window size m is chosen to be 1, 2, 3 and 4. There are at most 13 representations for a 12-layer transformer network. By setting window size to $m = 4$, we can cover a wide range of representations already. The performance versus the m value is given in Fig. 3 (a). As mentioned before, since the first several layers carry little contextualized information, it may not be necessary to include representations in the first several layers. We choose the starting layer l_S to be from 0-6 in the sensitivity study. The performance versus the l_S value is given in Fig. 3 (b). We see from both figures that SBERT-WK is robust to different values of m and l_S . By considering the performance and computational efficiency, we set window size $m = 2$ as the default value. For starting layer selection, the performance goes up a little bit when the representations of first three layers are excluded. This is especially true for the SST2 dataset. Therefore, we set $l_S = 4$ as the default value. These two default settings are used throughout all reported experiments in other subsections.

E. Inference Speed

We evaluate the inference speed against the STSB datasets. For fair comparison, the batch size is set to 1. All benchmarking methods are run on CPU and GPU⁶. Both results are reported. On the other hand, we report CPU results of

⁶Intel i7-5930K of 3.50GHz and Nvidia GeForce GTX TITAN X are chosen to be the CPU and the GPU, respectively.

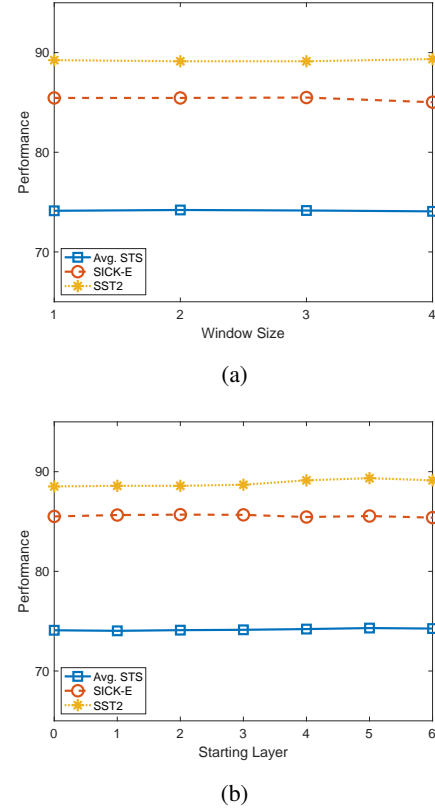


Fig. 3: Performance comparison with respect to (a) window size m and (b) starting layer l_S , where the performance for the STS dataset is the Pearson Correlation Coefficients (%) while the performance for the SICK-E and the SST2 datasets is test accuracy.

SBERT-WK only. All results are given in Table VIII. With CPU, the total inference time of SBERT-WK (QR) is 8.59 ms (overhead) plus 168.67ms (SBERT baseline). As compared with the baseline BERT model, the overhead is about 5%. SVD computation is slightly slower than QR factorization.

TABLE VIII: Inference time comparison of InferSent, BERT, XLNET, SBERT and SBERT-WK. Data are collected from 5 trails.

Model	CPU (ms)	GPU (ms)
InferSent	53.07	15.23
BERT	86.89	15.27
XLNET	112.49	20.98
SBERT	168.67	32.19
Overhead of SBERT-WK (SVD)	10.60	-
Overhead of SBERT-WK (QR)	8.59	-

VI. CONCLUSION AND FUTURE WORK

In this work, we provided in-depth study of the evolving pattern of word representations across layers in deep contextualized models. Furthermore, we proposed a novel sentence embedding model, called SBERT-WK, by dissecting deep contextualized models, leveraging the diverse information

learned in different layers for effective sentence representations. SBERT-WK is efficient, and it demands no further training. Evaluation was conducted on a wide range of tasks to show the effectiveness of SBERT-WK.

Based on this foundation, we may explore several new research topics in the future. Subspace analysis and geometric analysis are widely used in distributional semantics. Post-processing of the static word embedding spaces leads to furthermore improvements on downstream tasks [37], [62]. Deep contextualized models have achieved supreme performance in recent natural language processing tasks. It could be beneficial by incorporating subspace analysis in the deep contextualized models to regulate the training or fine-tuning process. This representation might yield even better results. Another topic is to understand deep contextualized neural models through subspace analysis. Although deep contextualized models achieve significant improvements, we still do not understand why these models are so effective. Existing work that attempts to explain BERT and the transformer architecture focuses on experimental evaluation. Theoretical analysis of the subspaces learned by deep contextualized models could be the key in revealing the myth.

ACKNOWLEDGMENT

Computation for the work was supported by the University of Southern California's Center for High Performance Computing (hpc.usc.edu).

REFERENCES

- [1] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.
- [2] 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)*, 2014, pp. 1532–1543.
- [3] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.
- [4] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. of NAACL*, 2018.
- [5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [6] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *Advances in neural information processing systems*, 2019, pp. 5754–5764.
- [7] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language_understanding_paper.pdf, 2018.
- [8] G. Jawahar, B. Sagot, D. Seddah, S. Unicomb, G. Iñiguez, M. Karsai, Y. Léo, M. Karsai, C. Sarraute, É. Fleury et al., "What does bert learn about the structure of language?" in *57th Annual Meeting of the Association for Computational Linguistics (ACL)*, Florence, Italy, 2019.
- [9] N. F. Liu, M. Gardner, Y. Belinkov, M. E. Peters, and N. A. Smith, "Linguistic knowledge and transferability of contextual representations," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.
- [10] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 353–355.
- [11] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.
- [12] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Advances in neural information processing systems*, 2015, pp. 3294–3302.
- [13] F. Hill, K. Cho, and A. Korhonen, "Learning distributed representations of sentences from unlabelled data," *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.
- [14] 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, September 2017, pp. 670–680.
- [15] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. St. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope, and R. Kurzweil, "Universal sentence encoder for English," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 169–174.
- [16] S. Subramanian, A. Trischler, Y. Bengio, and C. J. Pal, "Learning general purpose distributed sentence representations via large scale multi-task learning," *International Conference on Learning Representations*, 2018.
- [17] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, 2019.
- [18] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for SQuAD," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 784–789.
- [19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, E. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.
- [21] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *International Conference on Learning Representations*, 2019.
- [22] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, and S. Riedel, "Language models as knowledge bases?" *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.
- [23] O. Kovaleva, A. Romanov, A. Rogers, and A. Rumshisky, "Revealing the dark secrets of bert," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.
- [24] P. Michel, O. Levy, and G. Neubig, "Are sixteen heads really better than one?" in *Advances in Neural Information Processing Systems*, 2019, pp. 14 014–14 024.
- [25] Y. Hao, L. Dong, F. Wei, and K. Xu, "Visualizing and understanding the effectiveness of bert," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.
- [26] K. Ethayarajh, "How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.
- [27] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," *International Conference on Learning Representations (ICLR)*, 2017.
- [28] K. Ethayarajh, "Unsupervised random walk sentence embeddings: A strong but simple baseline," in *Proceedings of The Third Workshop on Representation Learning for NLP*, 2018, pp. 91–100.
- [29] Z. Yang, C. Zhu, and W. Chen, "Parameter-free sentence embedding via orthogonal basis," 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)*, 2019, pp. 638–648.

- [30] A. Rücklé, S. Eger, M. Peyrard, and I. Gurevych, "Concatenated power mean word embeddings as universal cross-lingual sentence representations," *arXiv preprint arXiv:1803.01400*, 2018.
- [31] V. Sanh, T. Wolf, and S. Ruder, "A hierarchical multi-task approach for learning embeddings from semantic tasks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6949–6956.
- [32] C.-C. J. Kuo, "Understanding convolutional neural networks with a mathematical model," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 406–413, 2016.
- [33] C.-C. J. Kuo, M. Zhang, S. Li, J. Duan, and Y. Chen, "Interpretable convolutional neural networks via feedforward design," *Journal of Visual Communication and Image Representation*, vol. 60, pp. 346–359, 2019.
- [34] A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard, "The robustness of deep networks: A geometrical perspective," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 50–62, 2017.
- [35] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse processes*, vol. 25, no. 2-3, pp. 259–284, 1998.
- [36] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Advances in neural information processing systems*, 2014, pp. 2177–2185.
- [37] B. Wang, F. Chen, A. Wang, and C.-C. J. Kuo, "Post-processing of word representations via variance normalization and dynamic embedding," *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 718–723, 2019.
- [38] Y. Tsvetkov, M. Faruqui, W. Ling, G. Lample, and C. Dyer, "Evaluation of word vector representations by subspace alignment," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 2049–2054.
- [39] B. Wang, A. Wang, F. Chen, Y. Wang, and C.-C. J. Kuo, "Evaluating word embedding models: Methods and experimental results," *APSIPA Transactions on Signal and Information Processing*, 2019.
- [40] D. Mekala, V. Gupta, B. Paranjape, and H. Karnick, "Scdv: Sparse composite document vectors using soft clustering over distributional representations," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 659–669.
- [41] N. Almarwani, H. Aldarmaki, and M. Diab, "Efficient sentence embedding using discrete cosine transform," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.
- [42] S. Kayal and G. Tsatsaronis, "Eigensent: Spectral sentence embeddings using higher-order dynamic mode decomposition," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 4536–4546.
- [43] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 1–14.
- [44] A. Conneau and D. Kiela, "Senteval: An evaluation toolkit for universal sentence representations," *arXiv preprint arXiv:1803.05449*, 2018.
- [45] E. Agirre, D. Cer, M. Diab, and A. Gonzalez-Agirre, "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)*, 2012, pp. 385–393.
- [46] E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo, "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*, 2013, pp. 32–43.
- [47] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, R. Mihalcea, G. Rigau, and J. Wiebe, "Semeval-2014 task 10: Multilingual semantic textual similarity," in *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, 2014, pp. 81–91.
- [48] E. Agirre, C. Banea, C. Cardie, D. Cer, M. Diab, A. Gonzalez-Agirre, W. Guo, I. Lopez-Gazpio, M. Maritxalar, R. Mihalcea et al., "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)*, 2015, pp. 252–263.
- [49] E. Agirre, C. Banea, D. Cer, M. Diab, A. Gonzalez-Agirre, R. Mihalcea, G. Rigau, and J. Wiebe, "Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation," in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 2016, pp. 497–511.
- [50] M. Marelli, S. Menini, M. Baroni, L. Bentivogli, R. Bernardi, R. Zamparelli et al., "A sick cure for the evaluation of compositional distributional semantic models," in *LREC*, 2014, pp. 216–223.
- [51] K. Ethayarajh, D. Duvenaud, and G. Hirst, "Towards understanding linear word analogies," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3253–3262.
- [52] S. Eger, A. Rücklé, and I. Gurevych, "Pitfalls in the evaluation of sentence embeddings," *4th Workshop on Representation Learning for NLP*, 2019.
- [53] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," in *Proceedings of the 43rd annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2005, pp. 115–124.
- [54] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 168–177.
- [55] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2004, p. 271.
- [56] J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language resources and evaluation*, vol. 39, no. 2-3, pp. 165–210, 2005.
- [57] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.
- [58] X. Li and D. Roth, "Learning question classifiers," in *Proceedings of the 19th international conference on Computational linguistics—Volume 1*. Association for Computational Linguistics, 2002, pp. 1–7.
- [59] B. Dolan, C. Quirk, and C. Brockett, "Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources," in *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, 2004, p. 350.
- [60] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni, "What you can cram into a single vector: Probing sentence embeddings for linguistic properties," *56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.
- [61] C. S. Perone, R. Silveira, and T. S. Paula, "Evaluation of sentence embeddings in downstream and linguistic probing tasks," *arXiv preprint arXiv:1806.06259*, 2018.
- [62] J. Mu, S. Bhat, and P. Viswanath, "All-but-the-top: Simple and effective postprocessing for word representations," *International Conference on Learning Representations*, 2018.



Bin Wang Bin Wang received his Bachelors degree in Electronic Information Engineering from University of Electronic Science and Technology of China (UESTC), Chengdu, China in June, 2017. Since July 2017, He joined Media Communication Lab (MCL) at University of Southern California (USC) as a Ph.D. student, supervised by Prof. C.-C. Jay Kuo. His research interests include natural language processing (NLP), image processing and machine learning.



C.-C. Jay Kuo C.-C. Jay Kuo (F99) received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1980, and the M.S. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1985 and 1987, respectively. He is currently the Director of the Multimedia Communications Laboratory and a Distinguished Professor of electrical engineering and computer science at the University of Southern California, Los Angeles. His research interests include multimedia computing and machine learning. Dr. Kuo is a Fellow of the American Association for the Advancement of Science (AAAS) and The International Society for Optical Engineers (SPIE).