Beyond Vectors: Subspace Representations for Embedding Set Operations

Anonymous ACL submission

Abstract

In natural language processing (NLP), the role of embeddings in representing linguistic semantics is crucial. Despite the prevalence of vector representations in embedding sets, they exhibit limitations in expressiveness and lack comprehensive set operations. To address this, we attempt to formulate and apply sets and their operations within pre-trained embedding spaces. Inspired by quantum logic, we propose to go beyond the conventional vector set representation with our novel subspace-based approach. This methodology constructs subspaces using pre-trained embedding sets, effectively preserving semantic nuances previously overlooked, and consequently consistently improving performance in downstream tasks. [1]

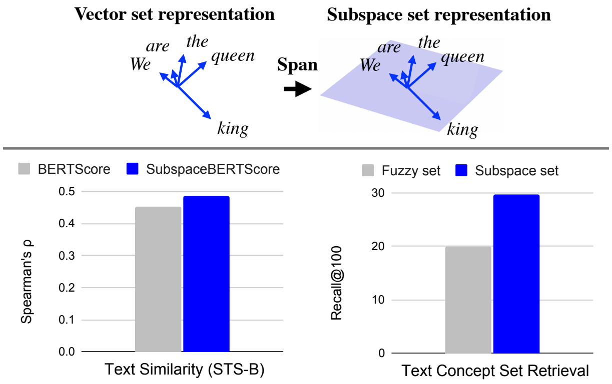


Figure 1: Superiority of subspace representations: Our subspace representation (blue) surpasses the traditional vector set representation (gray) in both text similarity and text concept set retrieval tasks.

1 Introduction

Embedding-based word representations have become fundamental in the field of natural language processing (NLP). Models like word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), along with recent Transformer-based architectures (Vaswani et al., 2017; Devlin et al., 2019), have underscored the significance of embeddings in capturing the complexities of linguistic semantics.

The importance of representing collections of words is pivotal in understanding concepts and relationships within language contexts (Zaheer et al., 2017; Zhelezniak et al., 2019). For instance, while words like "apple" and "orange" each carry their distinct meanings, together they represent the broader concept of fruits. Another example of important application is a sentence representation (Zaheer et al., 2017). The set of words in a sentence captures the overall meaning, allowing for computations such as text similarity (Agirre et al., 2012).

Against this backdrop, our research recognizes the significance of applying set operations in NLP and explores a new approach. Set operations enable a richer representation of relationships between collections of words, leading to more accurate semantic analysis based on context. For example, employing set operations allows for a clearer understanding of shared semantic features and differences among word groups within a text. This directly benefits tasks like determining semantic similarity and expanding word sets.

In response to these challenges, our study introduces a novel methodology that exploits the principles of *quantum logic* (Birkhoff and Von Neumann, 1936), applied within embedding spaces to define set operations. Our proposed framework adopts a subspace-based approach for representing word sets, aiming to maintain the intricate semantic relationships within these sets. We represent a word set as a subspace which is spanned by pre-trained embeddings. Additionally, it adheres to the foundational laws of set theory as delineated in the framework of quantum logic. This compliance ensures that our set operations, such as union, intersection, and complement, are not only mathematically robust but also linguistically meaningful

[1] Our codes will be publicly available after acceptance.

when applied them in pre-trained embedding space.

We first introduce a subspace set representation along with basic operations ($\cap$, $\cup$, and $\in$). Subsequently, to highlight the usefulness of our proposed framework, we introduce two core set computations: text similarity and set membership. The empirical results consistently point towards the notable superiority of our approach; our straightforward approach of spanning subspaces with pre-trained embedding sets enables a rich set representation, and we demonstrated its consistent performance enhancement in downstream tasks (Figure 1). Our research contributions include:

1. The introduction of continuous set representations and a framework for set operations, enabling more effective manipulation of word embedding sets (§ 4).

2. We propose *SubspaceBERTScore*, an extension of the embedding set-based text similarity method, BERTScore (Zhang et al., 2020). By simply transitioning from a vector set representation to a subspace, and incorporating a subspace-based indicator function, we observe a salient improvement in performance across all text similarity benchmarks (§ 5).

3. We apply subspace-based basic operations ($\cap, \cup$, and $\in$) to set expansion task and achive high performance (§ 6).

2 Preliminaries

To make the following discussion clear, we define several symbols. The sets of tokens in two sentences (A and B) are denoted as $A = \{a_1, a_2, \dots\}, B = \{b_1, b_2, \dots\}$ respectively. The sets of contextualized token vectors are denoted as $\mathbf{A} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \dots\}, \mathbf{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \dots\}$, where $\boldsymbol{a}$ and $\boldsymbol{b}$ are token vectors generated by the pre-trained embedding model such as BERT. The subspace spanned by $\mathbf{A}$ is denoted as $\mathbb{S}_A = \mathrm{span}(\boldsymbol{a}_1, \boldsymbol{a}_2, \dots)$. Note that the bases of the subspace is orthonormalized.

3 Symbolic Set Operations

We first formulate various set operations in a pre-trained embedding space. Among many types of operations for practical NLP applications, this work focuses on *set similarity*:

$$\begin{aligned} &A = \{A, \mathit{boy}, \mathit{walks}, \mathit{in}, \mathit{this}, \mathit{park}\}, \\ &B = \{\mathit{The}, \mathit{kid}, \mathit{runs}, \mathit{in}, \mathit{the}, \mathit{square}\}, \\ &\mathrm{Similarity}(A, B), \end{aligned} \tag{1}$$

set membership ($\in$) and *basic operations* ($\cap$, $\cup$):

$$\begin{aligned} &\mathit{Color} = \{\mathit{red}, \mathit{blue}, \mathit{green}, \mathit{orange}, \dots\}, \\ &\mathit{Fruit} = \{\mathit{apple}, \mathit{orange}, \mathit{peach}, \dots\}, \\ &\mathit{orange} \in \mathit{Color} \cap \mathit{Fruit}. \end{aligned} \tag{2}$$

For this purpose, we need following representations on a pre-trained embedding space[2]:

An element and a set of elements The representations of an element and a set of elements are the most basic ones. To exploit word embeddings, we represent a word (e.g., *orange*) as an element and a group of words (e.g., $\{\mathit{red}, \mathit{blue}, \mathit{green}, \mathit{orange}, \dots\}$) as a word set.

Quantification of set membership (indicator function) *Membership* denotes a relation in which word w is an element of set A, i.e., $w \in A$. We quantify it based on vector representations. Although the membership is typically a binary decision identical to that in a symbolic space, it can also be measured by the degree of closeness in a continuous vector space. Membership can be computed as an indicator function. The indicator function $\mathbb{1}_{\text{set}}$ quantifies whether the word w is included (1) or not (0) in the set in a discrete manner:

$$\mathbb{1}_{\text{set}}[w \in A] = \begin{cases} 1 & \text{if} \quad w \in A, \\ 0 & \text{if} \quad w \notin A. \end{cases} \tag{3}$$

Similarity between discrete symbol sets Set similarity, such as recall and precision, is an essential operation when calculating the similarity of texts. Despite its simplicity, the word overlap-based sentence similarity serves as a remarkably effective approximation and has found widespread practical application, as evidenced by numerous studies(Bojar et al., 2018; Zhang et al., 2020; Cer et al., 2017; Zhelezniak et al., 2019). They stand out as excellent similarity metrics based on embeddings. BERTScore (Zhang et al., 2020), which utilizes embeddings for its computation, is grounded

[2]These operations do not include some set operations such as cardinality, but are sufficient for expressing the practical forms of sets such as Eq. (1).

in recall and precision [3]. The typical computations for recall (R) and precision (P) are as follows[4]:

$$R = \frac{1}{|A|} \sum_{a_i \in A} \mathbb{1}_{\text{set}}[a_i \in B], \tag{4}$$

$$P = \frac{1}{|B|} \sum_{b_i \in B} \mathbb{1}_{\text{set}}[b_i \in A], \tag{5}$$

Basic set operations We need three basic set operations: *intersection* ($A \cap B$), *union* ($A \cup B$), and *complement* ($\overline{A}$). They allow us to represent various sets using such different combinations as $Color \cap Fruit$.

4 Subspace-based Set Representations

We propose the representations of a word set and set operations based on *quantum logic* (Birkhoff and Von Neumann, 1936). They hold advantages of geometric properties in an embedding space, and the set operations are guaranteed to hold for the laws of a set defined in quantum logic.

4.1 Quantum logic

While word embedding represents a word's meaning as a vector in linear space, quantum mechanics similarly represents a quantum state as a vector in linear space. These two intuitively different fields are very close to each other in terms of the representation and the operation of information.

Quantum logic (Birkhoff and Von Neumann, 1936) theory describes quantum mechanical phenomena. Intuitively, it is a framework for *set operations in a vector space*. In quantum logic, a set of vectors is represented as a linear subspace in a Hilbert space, and such set operations as union, intersection, and complement are defined as operations on subspaces. Quantum logic, which employs a complete orthomodular lattice as its system of truth values, guarantees to hold various set operations, such as De Morgan's laws $\overline{(A \cap B)} = \overline{A} \cup \overline{B}$ and $\overline{(A \cup B)} = \overline{A} \cap \overline{B}$, idempotent law: $A \cap A = A$, and double complement: $\overline{\overline{A}} = A$.

4.2 Set Operations in an Embedding Space

The representations of an element, a set, and such set operations as union, intersection, and complement in quantum logic can be applied directly in a word embedding space because it is a Euclidean space and therefore also a Hilbert space. However, since set similarity and set membership for a word embedding space are still missing in quantum logic, we propose a novel formulation of those operations using subspace-based representations, which is consistent with quantum logic. The correspondence between symbolic and subspace-based set operations is shown in Table 1.

[3] Unlike the symbolic set similarity, which do not consider word order, contextualized embeddings enable the capture of word sequence information.

[4] For simplicity in explanation, we present A and B in Eq. (4) and Eq. (5) as a set of tokens.

Algorithm 1 Computing basis of a subspace

Input: $\{\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(k)}\} \subseteq \mathbb{R}^{1\times d}$: Word embeddings to span subspace $\mathbb{S}_A$
Output: $\mathbf{S}_\mathrm{A} \in \mathbb{R}^{r\times d}$: Bases of $\mathbb{S}_A$
  $\mathbf{A} \in \mathbb{R}^{k\times d} \leftarrow \text{STACK\_ROWS}(\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(k)})$
  $\mathbf{S}_\mathrm{A} \in \mathbb{R}^{r\times d} \leftarrow (\text{ORTHO\_NORMAL}(\mathbf{A}^\top))^\top$ ▷ Orthonormalize the bases. r is the rank of $\mathbf{A}$
  return $\mathbf{S}_\mathrm{A}$

Set and elements Let $\mathbb{R}^n$ be a n-dimensional embedding space (Euclidean space), let $A = \{w_1, w_2, \ldots\}$ be a set of words, and let $\boldsymbol{v}_w \in \mathbb{R}^n$ be a word (token) vector corresponding to w. As discussed in §3, we first formulate the representation of a word and a word set. In quantum logic, an element is represented by a vector, and a set is represented by a subspace spanned by the vectors corresponding to its elements. Here we assume an element, i.e., word w, is represented by vector $\boldsymbol{v}_w$, and a word set is represented by linear subspace $\mathbb{S}_A \subset \mathbb{R}^n$ spanned by word vectors:

$$\mathbb{S}_A := \text{span}(\mathbf{A}) := \text{span}(\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots). \tag{6}$$

Hereinafter we simply refer to *linear subspace* as *subspace*. Algorithm 1 is the pseudocode for computing the basis of the subspace.

Basic set operations The *complement* of set A, denoted by $\overline{A}$, is represented by the orthogonal complement of subspace $\mathbb{S}_A$:

$$\mathbb{S}_{\overline{A}} := (\mathbb{S}_A)^\perp = \{\boldsymbol{v} \mid \exists \boldsymbol{a} \in \mathbb{S}_A, \boldsymbol{v} \cdot \boldsymbol{a} = 0\}. \tag{7}$$

The *union* of two sets, A and B, denoted by $A \cup B$, is represented by the sum space of two subspaces, $\mathbb{S}_A$ and $\mathbb{S}_B$:

$$\mathbb{S}_{A\cup B} := \mathbb{S}_A + \mathbb{S}_B = \{\boldsymbol{a} + \boldsymbol{b} \,|\, \boldsymbol{a} \in \mathbb{S}_A, \boldsymbol{b} \in \mathbb{S}_B\}. \tag{8}$$

The *intersection* of two sets, A and B, denoted by $A \cap B$, is represented by the intersection of two

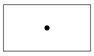
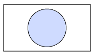
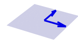
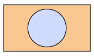
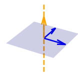
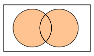
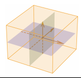
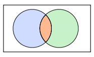
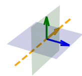
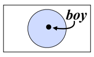
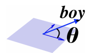
Symbolic Set Representation		Subspace-based Set Representation	
Element *king*		Vector $\boldsymbol{v}_{king}$	
Set $Male = \{king, man, \ldots\}$		Subspace $\mathbb{S}_{Male} = \mathrm{span}(\boldsymbol{v}_{king}, \boldsymbol{v}_{man}, \ldots)$	
Complement $\overline{Male}$		Orthogonal complement $(\mathbb{S}_{Male})^{\perp}$	
Union $Male \cup Female$		Sum space $\mathbb{S}_{Male} + \mathbb{S}_{Female}$	
Intersection $Color \cap Fruit$		Intersection $\mathbb{S}_{Color} \cap \mathbb{S}_{Fruit}$	
Set membership $boy \in Male$		Subspace indicator function $\mathbb{1}_{\mathrm{subspace}}(\boldsymbol{v}_{boy}, \mathbb{S}_{Male})$	

Table 1: Correspondence between symbolic set representations and subspace-based set representations: We demonstrate that union, intersection, and complement, which are formulated in quantum logic, and our new formulations of set membership and word set similarity hold in pre-trained word embedding space.

subspaces, $\mathbb{S}_A$ and $\mathbb{S}_B$:

$$\mathbb{S}_{A\cap B} := \mathbb{S}_A \cap \mathbb{S}_B = \{\boldsymbol{v} \mid \boldsymbol{v} \in \mathbb{S}_A, \boldsymbol{v} \in \mathbb{S}_B\}. \quad (9)$$

The basis of the intersection can be computed based on singular value decomposition (SVD). The bases are the vectors shared by the two subspaces.

Hard membership The set membership in the embedding space (e.g., $boy \in Male$) can be represented by the inclusion of a vector into a subspace (e.g., $\boldsymbol{v}_{boy} \in \mathbb{S}_{Male}$) and given by the following indicator function:

$$\mathbb{1}_{\mathrm{hard}}(\boldsymbol{v}, \mathbb{S}_A) := \begin{cases} 1 & (\boldsymbol{v} \in \mathbb{S}_A), \\ 0 & (\boldsymbol{v} \notin \mathbb{S}_A). \end{cases} \quad (10)$$

However, this binary decision fails to exploit the geometric properties of the word embedding space regarding semantic similarity. Suppose we quantify the degree of membership of word *boy* for word set $Male$ consisting of many masculine nouns other than boy. Even if $\boldsymbol{v}_{boy}$ is located very close to $\mathbb{S}_{Male}$ due to its semantic similarity to masculine nouns, $\mathbb{1}_{\mathrm{hard}}(\boldsymbol{v}_{boy}, \mathbb{S}_{Male})$ must return 0 because $\boldsymbol{v}_{boy}$ must not be located *exactly* on subspace $\mathbb{S}_{Male}$ defined by $Male$. It must return 1 based on the masculine property of word *boy*. Such *hard* membership defined by Eq. (10) is incompatible with an embedding space.

Soft membership: Subspace indicator function Instead, we define another membership function called *subspace indicator function* $\mathbb{1}_{\mathrm{subspace}}$ that returns continuous values from 0 to 1 depending on the following minimum angle between vector $\boldsymbol{v}_w$ and subspace $\mathbb{S}_A$ (the first canonical angle):

$$\mathbb{1}_{\mathrm{subspace}}(\boldsymbol{v}, \mathbb{S}_A) := \max\left\{ \frac{|\boldsymbol{a} \cdot \boldsymbol{v}|}{\|\boldsymbol{a}\|\|\boldsymbol{v}\|} \,\middle|\, \boldsymbol{a} \in \mathbb{S}_A \right\}. \quad (11)$$

This captures the degree of membership between a word and a word set, represented by the angle between a word vector and a subspace. It is a natural extension of $\mathbb{1}_{\mathrm{hard}}$, i.e., $\mathbb{1}_{\mathrm{subspace}}$ returns 1 when $\boldsymbol{v}_w \in \mathbb{S}_A$ and 0 when $\boldsymbol{v}_w \in \mathbb{S}_{\overline{A}}$.

The key distinction of our subspace indicator function approach lies in its ability to leverage the comprehensive information encapsulated within pre-trained word embedding space. The subspace indicator function does not simply find the nearest individual word from the set. Instead, we consider the *closeness* of the query word to the entire set as a whole, **by projecting the query word into the subspace spanned by the pre-trained embeddings** (as illustrated in the figure of the subspace indicator function function in Table 1). This way, we account not just for the individual word similarities, but also for the overall semantic coherence of the word set. The detailed process for computing this subspace indicator function can be found in Algorithm 2.

4.3 Set Similarity

Limitation of symbolic set similarities Suppose we quantify the set similarity between $A = \{A, boy,$

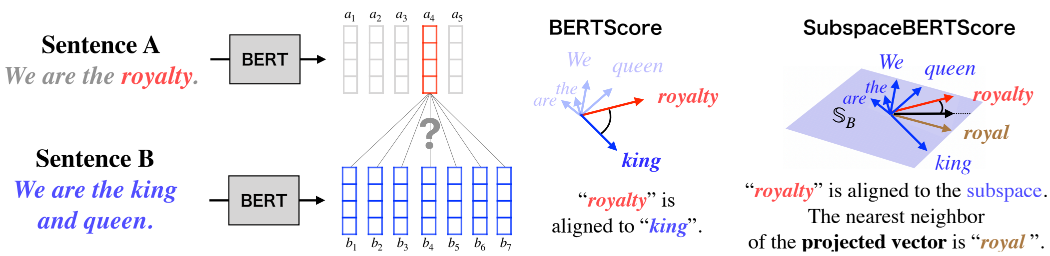


Figure 2: Comparison between the proposed SubspaceBERTScore and BERTScore. We visualize the alignment process between the word *royalty* and the words in the sentence B. SubspaceBERTScore represents B as the subspace $\mathbb{S}_B$ and calculates the similarity (canonical angle) between $\mathbb{S}_B$ and the *royalty* vector ($\boldsymbol{a}_4$). Our approach provides a "softer" alignment, capturing the overall semantic context of the sentence. On the other hand, BERTScore adopts a "harder" alignment strategy, selecting only the word from the sentence with the maximum cosine similarity.

walks, *in*, *this*, *park*$\}$ and $B = \{$*The*, *kid*, *runs*, *in*, *the*, *square*$\}$, which represent semantically similar sentences. The challenge with traditional symbolic set similarities, such as recall, is that they primarily rely on the exact overlap of words between the sets. These semantically similar sentences share only one word $\{in\}$ between A and B: $|A \cap B| = 1$. To address this shortcoming, it is essential to compute vector-based set similarity, such as BERTScore.

Three types of similarity in BERTScore BERTScore is a method that uses embeddings to approximately calculate R, P, and the F-score:

$$R_{\text{BERT}} = \frac{1}{|A|} \sum_{\boldsymbol{a}_i \in \mathbf{A}} \mathbb{1}_{\text{vectors}}(\boldsymbol{a}_i, \mathbf{B}), \tag{12}$$

$$P_{\text{BERT}} = \frac{1}{|B|} \sum_{\boldsymbol{b}_i \in \mathbf{B}} \mathbb{1}_{\text{vectors}}(\boldsymbol{b}_i, \mathbf{A}), \tag{13}$$

$$F_{\text{BERT}} = 2\frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}}, \tag{14}$$

where a sentence is represented as a set of token vectors $\mathbf{A}$ and $\mathbf{B}$. $\mathbb{1}_{\text{vectors}}$ is the indicator function for vector sets. Intuitively, this indicator function represents the calculation of *selecting one token from the sentence* and serves as a flexible extension of the binary indicator function $\mathbb{1}_{\text{set}}$. It returns a continuous similarity score between -1 and 1 for a token, depending on its similarity with the tokens in the sentence. Specifically, $\mathbb{1}_{\text{vectors}}(\boldsymbol{a}_i, \mathbf{B})$ quantifies to what extent the i-th token vector $\boldsymbol{a}_i$ in sentence A is semantically included in sentence B by taking the maximum cosine similarity between $\boldsymbol{a}_i$ and the token vectors in B:

$$\mathbb{1}_{\text{vectors}}(\boldsymbol{a}_i, \mathbf{B}) = \max_{\boldsymbol{b}_j \in \mathbf{B}} \cos(\boldsymbol{a}_i, \boldsymbol{b}_j) \in [-1, 1], \tag{15}$$

where $\cos(\boldsymbol{a}_i, \boldsymbol{b}_j)$ is the cosine similarity between $\boldsymbol{a}_i$ and $\boldsymbol{b}_j$.

SubspaceBERTScore To overcome the limitations of BERTScore regarding the expressiveness of its indicator function, we propose SubspaceBERTScore. This method extends BERTScore by employing the concept of subspace-based sentence representation and indicator function.

Extension of P, R, F with Subspaces Based on the above discussions, we propose *SubspaceBERTScore*, which calculates BERTScore's R, P, F using the subspace representation of sentences and the subspace indicator function:

$$R_{\text{subspace}} = \frac{1}{|A|} \sum_{\boldsymbol{a}_i \in A} \mathbb{1}_{\text{subspace}}(\boldsymbol{a}_i, \mathbb{S}_B), \tag{16}$$

$$P_{\text{subspace}} = \frac{1}{|B|} \sum_{\boldsymbol{b}_i \in B} \mathbb{1}_{\text{subspace}}(\boldsymbol{b}_i, \mathbb{S}_A), \tag{17}$$

$$F_{\text{subspace}} = 2\frac{P_{\text{subspace}} \cdot R_{\text{subspace}}}{P_{\text{subspace}} + R_{\text{subspace}}}, \tag{18}$$

where R_{subspace}, P_{subspace}, and F_{subspace} are the final evaluation measures of SubspaceBERTScore.

Weighting by Importance Previous study (Banerjee and Lavie, 2005; Vedantam et al., 2015) has shown that infrequently occurring words play a more important role in sentence similarity than general words. We apply importance weightings to our method as follows:

$$R_{\text{subspace}} = \frac{\sum_{\boldsymbol{a}_i \in A} \text{weight}(\boldsymbol{a}_i) \mathbb{1}_{\text{subspace}}(\boldsymbol{a}_i, \mathbb{S}_B)}{\sum_{\boldsymbol{a}_i \in A} \text{weight}(\boldsymbol{a}_i)}, \tag{19}$$

$$P_{\text{subspace}} = \frac{\sum_{\boldsymbol{b}_i \in B} \text{weight}(\boldsymbol{b}_i) \mathbb{1}_{\text{subspace}}(\boldsymbol{b}_i, \mathbb{S}_A)}{\sum_{\boldsymbol{b}_i \in B} \text{weight}(\boldsymbol{b}_i)}, \tag{20}$$

where $\text{weight}(\cdot)$ is a weighting function. We use the L2 norm of the vector (Yokoi et al., 2020).

Algorithm 2 Subspace indicator function $\mathbb{1}_{\text{subspace}}(\boldsymbol{v}_w, \mathbb{S}_A)$

Input: $\mathbf{S}_{\mathrm{A}} \in \mathbb{R}^{k \times d}$: Bases of $\mathbb{S}_A$
Input: $\boldsymbol{v}_w \in \mathbb{R}^{1 \times d}$: A word vector
Output: $\sigma \in \mathbb{R}$: Membership degree
  if $k = 0$ **then**
    return 0
  else
    $\widetilde{\boldsymbol{v}}_w \leftarrow \frac{\boldsymbol{v}_w}{||\boldsymbol{v}_w||} \in \mathbb{R}^{1 \times d}$
    $\mathbf{U}^\top \in \mathbb{R}^{k \times k}, \sigma \in \mathbb{R}, \mathbf{V} \in \mathbb{R} \leftarrow \mathrm{SVD}(\mathbf{S}_{\mathrm{A}} \widetilde{\boldsymbol{v}}_w^\top)$
    return $\sigma \in \mathbb{R}$ ▷ The output of $\mathbb{1}_{\text{subspace}}(\boldsymbol{v}_w, \mathbb{S}_A)$ is always **non-negative** because it is a singular value
  end if

5 Semantic Textual Similarity Task

In this section, we examine the effectiveness of SubspaceBERTScore through the semantic textual similarity task (STS; Agirre et al., 2012).

Task An STS task calculates the similarity between two sentences. For the STS evaluation protocol, we follow Gao et al. (2021). Its evaluation is based on the correlation between the similarity calculated by the model and corresponding human judgments. We used datasets from the SemEval shared task 2012-2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016), STS benchmark (STS-B; Cer et al., 2017), and SICK-Relatedness (SICK-R; Marelli et al., 2014). We used Spearman's ρ.

Embeddings We used 768-dimensional $\text{BERT}_{\text{base}}$[5] (Devlin et al., 2019), which was pre-trained with BookCorpus and Wikipedia. We used hidden states in the last layer.

Baselines We compared our method **SubspaceBERTScore** with other baseline similarity metrics. The baselines included **Avg-cos** (Arora et al., 2017), the cosine similarity between the averaged vectors, **CLS-cos** (Gao et al., 2021), the cosine similarity between the $[CLS]$ representations of the pre-trained language model, **DynaMax** (Zhelezniak et al., 2019), a set similarity based on fuzzy sets, Word Mover's Distance (**WMD**; Kusner et al., 2015), a metric based on optimal transport cost, and Word Rotator's Distance (**WMD**; Yokoi et al., 2020), an optimal transport-based metric that improves WMD.

[5] https://huggingface.co/bert-base-uncased

Main results The results are shown in Table 2. In comparison to BERTScore, our method achieves superior correlation with human judgments across all three key metrics: F-score, precision, and recall. An important observation is that the performance consistently improves by subspace representation of the set. The results suggest that simply replacing the representation of embedding sets and the indicator function with subspace-based alternatives significantly enhances our ability to capture and express the depth of linguistic semantics.

We also conduct an experiment using L2 norm as a weighting factor for the indicator function. This method has previously been proven effective in the STS task (Yokoi et al., 2020). We see that both our proposed method and BERTScore improve their performance underlining the effectiveness of this weighting approach in both cases. Notably, our proposed method continued to outperform BERTScore even when L2 norm was used for weighting.

Our similarity also outperforms the fuzzy-set based similarity of DynaMax. This result suggests that the proposed subspace-based approach represents a set and set operations better than the fuzzy set-based approach in embedding space.

6 Text Concept Set Retrieval Task

In this section, we evaluate the capability of our proposed set operations ($\cap$, $\cup$, and $\in$) in effectively representing word sets.

Task We evaluate our set operations by the set expansion task introduced by Zaheer et al. (2017). In this task, the model is given a set of words that share a common concept or theme. The objective is to expand this set by retrieving relevant words from a vocabulary that fit the same concept. For instance, if the initial set includes words like "apple", "banana", and "peach", the task would be to identify and add other fruit names (e.g., "orange") to this set. For the evaluation, we follow Zaheer et al. (2017). We report recall (**R@k**) and **Median**, that indicate whether the words in the test set can be ranked higher.

Embeddings We used the most standard pre-trained word embeddings in all of our experiments: 300-dimensional GloVe[6] (Pennington et al., 2014), which was pre-trained with Common Crawl, and 300-dimensional word2vec[7] (Mikolov et al., 2013),

[6] https://nlp.stanford.edu/projects/glove/
[7] https://code.google.com/archive/p/word2vec/

Method	Metric	Weighting	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
CLS-cos	-	-	.215	.321	.213	.379	.442	.203	.424	.314
Avg-cos	-	-	.309	.599★	.477	.603	.637	.473	.582★	.526
WMD	-	-	.238	.443	.389	.531	.532	.384	.509	.432
WRD	-	-	.241	.502	.410	.573	.573	.421	.527	.464
DynaMax	-	-	.322	.518	.432	.616	.639	.452	.560	.506
BERTScore	F	-	.312	.546	.450	.602	.636	.446	.553	.506
	P	-	.261	.532	.462	.576	.622	.443	.559	.494
	R	-	.350	.527	.416	.602	.623	.430	.522	.496
SubspaceBERTScore	F	-	**.335**	**.573**	**.476**	**.610**	**.650**	**.479**	**.562**	**.526**
	P	-	**.282**	**.550**	**.488**	**.580**	**.630**	**.475**	**.568**	**.511**
	R	-	**.369★**	**.552**	**.436**	**.611**	**.639**	**.462**	**.530**	**.514**
BERTScore	F	L2	.321	.540	.452	.613	.640	.454	.558	.511
	P	L2	.274	.529	.468	.589	.627	.450	.565	.500
	R	L2	.348	.520	.414	.610	.624	.437	.524	.497
SubspaceBERTScore	F	L2	**.342**	**.568**	**.477**	**.621**	**.653★**	**.486★**	**.568**	**.531★**
	P	L2	**.292**	**.547**	**.492★**	**.592**	**.634**	**.479**	**.574**	**.516**
	R	L2	**.367**	**.544**	**.434**	**.620★**	**.640**	**.468**	**.532**	**.515**

Table 2: A comprehensive comparison of similarity metrics in the STS task. The scores are Spearman's ρ. The methods with the highest values, using the same pre-trained embeddings, are highlighted in ★. Scores that showed improvement from BERTScore are denoted in **bold**.

which was pre-trained with Google News.

Set Expansion with Subspace Indicator Function To illustrate our subspace-based set expansion method (**Subspace Set**), we consider a set of fruit-related words. For example, let's take $S_{\text{fruit}} = \{apple, banana, \dots\}$. This set is divided into two subsets: a 'span' subset used for creating a subspace representation, and a 'test' subset for evaluation. Let's assume $orange \notin S_{\text{fruit\_span}}$ is a target word for testing. (1) From the 'span' subset, we generate a subspace: $\mathbb{S}_{Fruit} = \text{span}(S_{\text{fruit\_span}})$. For instance, if $S_{\text{fruit\_span}} = \{apple, banana, \dots\}$, then $\mathbb{S}_{Fruit} = \text{span}(\boldsymbol{v}_{apple}, \boldsymbol{v}_{banana}, \dots)$. (2) We define a subspace indicator function, which computes the degree to which a word vector belongs to the subspace. For a word w, the membership score is calculated as score $= \mathbb{1}_{\text{subspace}}(\boldsymbol{v}_{orange}, \mathbb{S}_{Fruit})$. This score reflects the extent to which w aligns with the semantic characteristics of the subspace. This method effectively expand the set S_{fruit} by identifying words that share semantic properties with the subspace defined by the initial set.

Baselines We compared several baselines, which don't require training on word sets, to our method. **Random** just selects words randomly from the dataset's vocabulary. A simple unsupervised baseline with word embeddings uses the nearest neighbors in the embedding space (**Near**)[8]. We also compare a method based on fuzzy sets (**Fuzzy set**; Zhelezniak et al., 2019) with our method. Similar to our method, their method is designed to exploit both the flexibility of word vectors and rich set operations. Fuzzy Set represents word set A by max-pooled word vectors $\boldsymbol{s} = \max_{w \in A} \boldsymbol{v}_w$. One major difference from our method is that Fuzzy set represents a set of word vectors by compressing them into a vector of fixed dimensions. Although the Text Concept Set Retrieval task requires computing the degree of a word's membership for a word set, their method does not provide it. We instead used cosine similarity $\cos(\boldsymbol{v}_w, \boldsymbol{s})$ between word vector $\boldsymbol{v}_w$ of word $w \in V$ and $\boldsymbol{s}$ as the degree of membership to apply fuzzy sets to the task.

Dataset We used a previously created dataset (Zaheer et al., 2017), which was denoted by "LDA-1k, Vocab = 17k." in the paper. The dataset ($\mathbf{D}^{\text{Set}}$) contains 100 word sets, each of which consists of 50 words sampled from a common topic[9]. Five pre-determined words from each set were used as the word set S. An additional 800 word sets were used to train the models that require training on word sets. Table 3 shows an example of the data and the number of test sets.

To evaluate the union and intersection sets, we prepared additional data through the union and

[8]While Zaheer et al. (2017) does not provide details about this method, we have inferred through our replication experiments that it uses a method based on the cosine similarity between the query word vector and other vectors to obtain the nearest neighbor.

[9]This work used Latent Dirichlet Allocation (LDA; Blei et al., 2003) as a topic model.

Dataset (# Set)	Example				
	Set	Words (set elements)			
$\mathbf{D}^{\text{Set}}$ (100)	S_3	daily	news	paper	...
$\mathbf{D}^{\text{Union}}$ (100)	S_{12}	rider	bike	bicycle	...
	S_{51}	island	fishing	sea	...
	$S_{12} \cup S_{51}$	races	cycling	islands	...
$\mathbf{D}^{\text{Intersect}}$ (100)	S_9	tour	open	golf	...
	S_{72}	poker	casino	gambling	...
	$S_9 \cap S_{72}$	money	won	player	...

Table 3: Examples from original dataset (denoted as $\mathbf{D}^{\text{Set}}$) and additional $\mathbf{D}^{\text{Union}}$ and $\mathbf{D}^{\text{Intersect}}$ sets.

	Method	Emb.	Set	R@100	R@1k	Med.
$\mathbf{D}^{\text{Set}}$	Rand♠	-	×	0.6	5.9	8520
	Near♠	w2v	×	28.1	54.7	641
	Fuzzy set	w2v	✓	19.9	47.2	1240
	Fuzzy set	GloVe	✓	30.9	69.0	320
	Subspace set	w2v	✓	29.7	58.9	478
	Subspace set	GloVe	✓	**35.7**	**72.7**	**246**
$\mathbf{D}^{\text{Union}}$	Rand	-	×	0.6	6.0	8422
	Near	w2v	×	17.5	34.3	3270
	Fuzzy set	w2v	✓	2.8	17.1	4426
	Fuzzy set	GloVe	✓	5.4	32.0	2347
	Subspace set	w2v	✓	18.4	46.9	1202
	Subspace set	GloVe	✓	**24.4**	**68.3**	**407**
$\mathbf{D}^{\text{Intersect}}$	Rand	-	×	0.2	6.6	7929
	Near	w2v	×	23.5	40.8	3304
	Fuzzy set	w2v	✓	4.7	20.9	3420
	Fuzzy set	GloVe	✓	32.5	75.0	255
	Subspace set	w2v	✓	25.7	45.7	1445
	Subspace set	GloVe	✓	**44.2**	**83.7**	**149**

Table 4: Results of set retrieval task on $\mathbf{D}^{\text{Union}}$ (top half) and $\mathbf{D}^{\text{Intersect}}$ (bottom half). The "Emb" column indicates which pre-trained embedding is used. The "Set" column indicates whether each method is based on set computations: ✓ for incorporating set operations.

intersection operations on two randomly-selected word sets from the original word sets ($\mathbf{D}^{\text{Set}}$)[10]. The number of words in each set in $\mathbf{D}^{\text{Union}}$ was limited to 50 to match the original dataset ($\mathbf{D}^{\text{Set}}$). The number of words in each set in $\mathbf{D}^{\text{Intersect}}$ was set to a minimum of 10. Finally, 100 unions and intersections were randomly selected from these word sets with zero elements excluded. See Table 3 for examples and statistics of the datasets.

Results In experiments on union and intersection, we compared our method only with Fuzzy Set. The proposed method and Fuzzy Set can induce representations for the union and intersection using set operations defined in the word embedding space; the others cannot do so directly. Table 4 shows the experimental results. Here our subspace-based set operation method (Subspace set) is the best among the methods that did not require training. The results suggest that combining off-the-shelf pre-trained embeddings with appropriate set-oriented operations makes linguistic computation on sets feasible without additional training. The results in $\mathbf{D}^{\text{Union}}$ and $\mathbf{D}^{\text{Intersect}}$ show that the our method outperform Fuzzy Set in most metrics. As methods for achieving set operations in vector spaces, the proposed method is empirically more promising than the existing fuzzy set-based method.

7 Related Work

Symbol-based similarities between word sets have been proposed, such as Jaccard coefficient (Jaccard, 1901; Manning and Schütze, 2001; Thada and Jaglan, 2013) and TF-IDF-based similarity (Jurafsky, 2000). Unfortunately, symbol-based methods cannot capture the semantic similarity of similar sets or words when the symbols are different.

While many studies have explored representing word sets in pre-trained embedding spaces (Kusner et al., 2015; Yokoi et al., 2020), they primarily focus on set similarity. Our approach, however, extends beyond this by developing a comprehensive framework for various set operations within these spaces. Utilizing subspace properties, our method not only represents word sets but also performs a range of versatile operations, such as calculating textual similarities and membership degrees.

Many methods for learning the representation of sets have been proposed because of the wide range of possible applications (Zaheer et al., 2017; Pellegrini et al., 2021; Lee et al., 2019; Vilnis and McCallum, 2015; Athiwaratkun and Wilson, 2017). In contrast, our approach does not require additional training. This enables us to compute set representations and operations using popular general-purpose language models, which are trained on the general domain (Brown et al., 2020).

8 Conclusion

This study introduces a novel framework for set representation and operations within pre-trained embedding spaces, employing linear subspaces grounded in quantum logic. This approach extends the scope of conventional embedding set operations by incorporating vector-based representations.

[10]Note that these were based on the dataset LDA-1k, which was automatically generated by Zaheer et al. (2017) using LDA, so the quality depends on their method.

Ethical Considerations

We recognize the importance of addressing the inherent biases in pre-trained models, such as gender stereotypes. In our experiment, we used RoBERTa, which has gender biases (Sharma et al., 2021). We used this model in its original state to preserve the experimental conditions of BERTScore, acknowledging that such biases may influence our results. However, we would like to emphasize that the focus of our work, which lies in sentence similarity, does not inherently add to or magnify these ethical concerns.

Limitations

Our SubspaceBERTScore is built upon the foundation of BERTScore, which presents a limitation in that our results and findings are inherently dependent on the characteristics and performance of BERTScore. While we chose BERTScore due to its robustness and popularity in the field, potential biases or shortcomings intrinsic to BERTScore might be incorporated into our extension. Nevertheless, this constraint also suggests future research possibilities, such as applying our subspace-based approach to other base sentence similarity metrics, further expanding the versatility and applicability of our method.

The experiments we conducted were exclusive to BERT and RoBERTa. Testing our methodology with other pre-trained models, like GPT-3 (Brown et al., 2020), could broaden its applicability and establish its robustness across various pre-trained models.

We evaluated our methodology primarily using English datasets. This decision was made to streamline our initial explorations rather than due to an inherent language-specific bias in our approach. We expect that our subspace-based methodology will be effective across various languages.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. 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@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 252–263. The Association for Computer Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. 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@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 81–91. The Association for Computer Linguistics.

Eneko Agirre, Carmen Banea, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 497–511. The Association for Computer Linguistics.

Eneko Agirre, Daniel M. Cer, Mona T. Diab, and Aitor Gonzalez-Agirre. 2012. SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012*, pages 385–393. The Association for Computer Linguistics.

Eneko Agirre, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *SEM 2013 shared task: Semantic Textual Similarity. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics, *SEM 2013, June 13-14, 2013, Atlanta, Georgia, USA*, pages 32–43. Association for Computational Linguistics.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Ben Athiwaratkun and Andrew Gordon Wilson. 2017. Multimodal Word Distributions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1645–1656.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization@ACL 2005, Ann Arbor, Michigan, USA, June 29, 2005*, pages 65–72. Association for Computational Linguistics.

Garrett Birkhoff and John Von Neumann. 1936. The logic of quantum mechanics. *Annals of mathematics*, pages 823–843.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Ond$v_r$ej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Daniel M. Cer, Mona T. 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@ACL 2017, Vancouver, Canada, August 3-4, 2017*, pages 1–14. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. Association for Computational Linguistics.

Paul Jaccard. 1901. Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bulletin de la Societe Vaudoise des Sciences Naturelles*, 37:241–72.

Dan Jurafsky. 2000. *Speech & language processing*. Pearson Education India.

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. 2015. From Word Embeddings To Document Distances. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 957–966. JMLR.org.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3744–3753. PMLR.

Christopher D. Manning and Hinrich Schütze. 2001. *Foundations of statistical natural language processing*. MIT Press.

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 2014, Reykjavik, Iceland, May 26-31, 2014*, pages 216–223. European Language Resources Association (ELRA).

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Giovanni Pellegrini, Alessandro Tibo, Paolo Frasconi, Andrea Passerini, and Manfred Jaeger. 2021. Learning Aggregation Functions. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 2892–2898. ijcai.org.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543.

Shanya Sharma, Manan Dey, and Koustuv Sinha. 2021. Evaluating gender bias in natural language inference. *CoRR*, abs/2105.05541.

Vikas Thada and Vivek Jaglan. 2013. Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm. *International Journal of Innovations in Engineering and Technology*, 2(4):202–205.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural*

Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 5998–6008.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575. IEEE Computer Society.

Luke Vilnis and Andrew McCallum. 2015. Word Representations via Gaussian Embedding. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Sho Yokoi, Ryo Takahashi, Reina Akama, Jun Suzuki, and Kentaro Inui. 2020. Word Rotator's Distance. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2944–2960. Association for Computational Linguistics.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. 2017. Deep Sets. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3391–3401.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Vitalii Zhelezniak, Aleksandar Savkov, April Shen, Francesco Moramarco, Jack Flann, and Nils Y. Hammerla. 2019. Don't Settle for Average, Go for the Max: Fuzzy Sets and Max-Pooled Word Vectors. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.