Static Fuzzy Bag-of-Words: Exploring Static Universe Matrices for Sentence Embeddings

Matteo Muffo, Roberto Tedesco, Licia Sbattella, and Vincenzo Scotti

Abstract Vector semantics has slightly become a key tool for natural language processing, especially concerning text analysis. This kind of vector representation is usually encoded through embeddings that can be used to encode semantic information at different levels of granularity. In fact, through the years, not only models for word embeddings have been developed but also for sentence and documents. With this work, we address *sentence embeddings*, in particular the *non-parametric* ones, which offer a good trade-off between performance and inference speed. We present *Static Fuzzy Bag-of-Word* (SFBoW) model, a refinement of the Fuzzy Bag-of-Words approach yielding fixed-dimension sentence embeddings. We targeted fixed-size embeddings to promote caching a re-usability, speeding the inference of a system that relies on our model. In this paper, we explore various approaches for the construction of a *static universe matrix*, fundamental to make the sentence embeddings of fixed size. To show the validity of our approach, we benchmarked our model on a semantic similarity task, obtaining competitive performances.

1 Introduction

The advent of machine and deep learning-based models has influenced many areas of the artificial intelligence field [10, 17], including natural language processing (NLP). To enable the deep neural network models, which strongly rely on matrix multiplication operations, process data from NLP, vector semantics has played a

M. Muffo (✉)
Indigo.ai, Milano, Italy
e-mail: matteo@indigo.ai

R. Tedesco · L. Sbattella · V. Scotti
Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, Milano, Italy
e-mail: roberto.tedesco@polimi.it; licia.sbattella@polimi.it; vincenzo.scotti@polimi.it

M. Abbas (ed.), *Analysis and Application of Natural Language and Speech Processing*, Signals and Communication Technology,
https://doi.org/10.1007/978-3-031-11035-1_10

crucial role. In fact, vector semantics rely on the concept that everything can be represented as real-valued vectors (or points) in a hyperspace. Moreover, according to vector semantics, the position of the object in the hyperspace represents its meaning.

In the case of NLP, words with similar meaning should be represented close in the hyperspace, and, analogously, words with different meaning should be far one from the other. This approach to word vector representation is called word embedding. These embeddings are computed through *self-supervised* representation learning [9]. There are many different models to extract these embeddings, at different levels of granularity, which have consistently taken the role of input representation for many NLP tasks [19].

In the last decade, the approach shifted from *shallow and static* word representations [11, 22, 25] towards *deep and contextual* ones [15, 26, 28], pushing forward incredibly the state of the art on NLP. However, for certain problems like web search and question answering, word-level representations are not sufficient. For this reason high-level models such as those for *sentence embedding* have been created [40]. These high-level models can be powered through either satic or contextual representations.

Shallow *word embedding* models immediately provided noticeable results [14]. Such representations were quickly adopted to provide an input for syntactic analysis: they helped improve results in part-of-speech (POS) tagging, named entity recognition (NER) and semantic role labelling (SRL). Shortly after, they were employed in more complex problems like language modelling, machine translation [36] and dialogue systems [35]. Although impressive, the results of these models were limited by the inability to model properly the context surrounding each word in the input sequence.

Neural language models (LMs) implemented through *transformer networks* [18, 37], on the other side, played a significant role for deep contextual representations. The hidden representations extracted through these huge models, trained on massive collections of unlabelled textual data, boosted the performances in many NLP tasks [29, 30, 38, 39] The trade-off with respect to shallow model is indeed in the amount of computational resources: both in terms of time and memory. This resource demand is especially high at train time.

In this vector semantics settings, with focus on the sentence embeddings, we present our Static Fuzzy Bag-of-Words (SFBoW) model. It's a model for non-parametric sentence embeddings based on the DynaMax Fuzzy Bag-of-Words model [42]. In particular, with this paper, we explore approaches to build the universe matrix, core component of the Fuzzy Bag-of-Words solutions, to be static. This model is designed to promote caching (in the sense of re-usability of the embeddings), short analysis time and valid performances; thus, it is advised for applications with limited resources or with power consumption issues, like embedded systems. To evaluate the goodness of the proposed universe matrices, we relied on the semantic textual similarity (STS) benchmark.

We organise the remainder of this paper into the following sections: in Sect. 2, we summarise the main concepts related to learnt word and sentence representations; in

Sect. 3, we introduce SFBoW, our model; in Sects. 4 and 5, we present, respectively, the evaluation approach we followed to evaluate SFBoW and the results of such evaluation; and, finally, in Sect. 6, we summarise the presented work and we present the expected future works.

2 Related Work

Our work revolves around the concept of *vector semantics*: the idea that the meaning of a word or a sentence can be modelled as a vector [23].

The first steps on this subject were made in information retrieval (IR) context with the vector space model [33], where documents and queries were represented as high-dimensional (vocabulary size) sparse embedding vectors. In this model, each dimension is used to represent a word, so that given a vocabulary $\mathcal{V}$:

- A word $w_i \in \mathcal{V}$, with $i \in [1, |\mathcal{V}|] \subseteq \mathbb{N}$, is expressed as a so-called "one-hot" binary vector $\mathbf{v}_{w_i} \in \mathbb{1}^{|\mathcal{V}|}$, where, calling $v_{w_i,j}$ the jth element of the word vector, it holds that $v_{w_i,j} = 1 \iff j = i$.
- A sentence S is expressed as vector $\boldsymbol{\mu}_S \in \mathbb{N}^{|\mathcal{V}|}$, where $\mu_{S,i}$, the ith element of vector $\boldsymbol{\mu}_S$, namely, $c_{S,i}$, represents the number of times word w_i appears in sentence S.

The resulting sentence representation, used also for text documents, is called *Bag-of-Words* (BoW) and can be summarised as

$$\boldsymbol{\mu}_S = \sum_{i=1}^{|\mathcal{V}|} c_{S,i} \cdot \mathbf{v}_{w_i}. \tag{1}$$

These representation models needed to be replaced because of the sparsity, which made them resource consuming, and the induced orthogonality among vectors with similar meanings.

2.1 Word and Sentence Embeddings

Word embeddings refer to the dense semantic vector representation of words; such representation can be divided into *prediction-based* and *count-based* [8].

The former group identifies the embeddings obtained through the training of models for next/missing word prediction given a context. It encompasses models like *Word2Vec* [21, 22] and *fastText* [11]. The latter group refers to the embeddings obtained leveraging word co-occurrence counts in a corpus. One of the most recent solutions of this group is *GloVe* [25].

All the models mentioned above belong to the class of *shallow* models, where the embedding of a word w_i can be extracted through lookup over the rows of the embedding matrix $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}| \times d}$, with d being the desired dimensionality of the embedding space. Given the word (column) vector $\mathbf{v}_{w_i}$, the corresponding word embedding $\mathbf{u}_{w_i} \in \mathbb{R}^d$ can be computed as (see Sect. 2.2)

$$\mathbf{u}_{w_i} = \mathbf{W}^\top \cdot \mathbf{v}_{w_i}. \tag{2}$$

More recently, the introduction of transformer-based LMs [18], like *BERT* [15], *GPT* [12, 26, 27] or *T5* [28], has spread the concept of *contextual embeddings*; such embeddings proved to be particularly helpful for a wide variety of NLP problems, as shown by the leader boards of NLP benchmarks [29, 30, 38, 39].

The inherent hierarchical structure of the human language makes it hard to understand a text from single words; thus, the birth of higher-level semantic representations for sentences, which are the sentence embeddings, was just a natural consequence. As for the word embeddings, also sentence embeddings are organised into two groups, *parametrised* and *non-parametrised*, depending on whether the model requires parameter training or not.

Clear examples of parametric model are the *skip-thoughts vectors* [16] and *Sent2Vec* [24], which generalises Word2Vec. Non-parametric models, instead, show that simply aggregating the information from pre-trained word embeddings, for example, through averaging, as in *SIF weighting* [6], is sufficient to represent higher-level entities like sentences and paragraphs.

Transformer LMs are also usable at sentence level. An example is the parametric model *Sentence-BERT* [31], obtained by fine-tuning on natural language inference corpora.

All these models rely on the assumption that cosine similarity is the correct metric to compute "meaning distance" between sentences. This is why parametric models are explicitly trained to minimise this measure for similar sentences and maximise it for dissimilar sentences.

However, cosine similarity may not be the only and best measure. The *DynaMax* model [42] proposed to follow a fuzzy set representation of sentences and to rely on fuzzy Jaccard similarity instead of the cosine one. As a result, the DynaMax model outperformed many non-parametric models and performed comparably to parametric ones under cosine similarity measurements, even if competitors were trained directly to optimise that metric, while the DynaMax approach was utterly unrelated to that objective.

The use of fuzzy sets to represent documents is not new, and it was already proposed by [41]. With respect to DynaMax, previous results were inferior because of their approach to compute fuzzy membership.

2.2 *Fuzzy Bag-of-Words and DynaMax for Sentence Embeddings*

The *Fuzzy Bag-of-Words* (FBoW) model for text representation [41]—and its generalised and improved variant DynaMax [42], which introduced a better similarity metric—represents the starting point of our work, which is described in Sect. 3.

The BoW approach, described at the beginning of Sect. 2, can be seen as a multi-set representation of text. It enables to measure similarity between two sentences with set similarity measures, like Jaccard, Otsuka and Dice indexes. These indexes share a common pattern to measure the similarity σ between two sets A and B [42]:

$$\sigma\left(A, B\right) = n_{shared}\left(A, B\right) / n_{total}\left(A, B\right) \tag{3}$$

where $n_{shared}\left(A, B\right)$ denotes the count of shared elements and $n_{total}\left(A, B\right)$ is the count of total elements. In particular, the Jaccard index is defined as

$$\sigma_{Jaccard}\left(A, B\right) = \left|A \cap B\right| / \left|A \cup B\right|. \tag{4}$$

However, the simple set similarity is a rigid approach as it allows for some degree of similarity when the very same words appear in both sentences, but fails in the presence of synonyms. This is where *fuzzy sets theory* comes handy: in fact, fuzzy sets enable to interpret each word in $\mathcal{V}$ as a singleton and measure the degree of membership of any word to this singleton as the similarity between the two considered words [41].

The FBoW model prescribes to work in this way [41]:

- Each word w_i is interpreted as a singleton $\{w_i\}$; thus, the membership degree of any word w_j in the vocabulary (with $j \in [1, |\mathcal{V}|] \subseteq \mathbb{N}$) with respect to this set is computed as the similarity σ between w_i and w_j. These similarities can be used to fill a $|\mathcal{V}|$-sized vector $\hat{\mathbf{v}}_{w_i}$ used to provide the fuzzy representation of w_i (the jth element $\hat{\mathbf{v}}_{w_i,j}$ being $\sigma\left(w_i, w_j\right)$).
- A sentence S is simply defined through the fuzzy union operator, which is determined by the max operator over the membership degrees. In this case, the S is represented by a vector of $|\mathcal{V}|$ elements.

The generalised FBoW approach [42] prescribes to compute the *fuzzy embedding* of a word singleton as

$$\hat{\mathbf{v}}_{w_i} = \mathbf{U} \cdot \mathbf{u}_{w_i} = \mathbf{U} \cdot \mathbf{W}^{\top} \cdot \mathbf{v}_{w_i} \tag{5}$$

to reduce the dimension of the output vector for S, where $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is a word embedding matrix (defined as in Sect. 2.1), $\mathbf{u}_{w_i}$ is defined in Eq. (2) and $\mathbf{U} \in \mathbb{R}^{u \times d}$ (with u being the desired dimension of the fuzzy embeddings) is the *universe matrix*, derived from the *universe set* U, which is defined as "the set of all possible terms that

occur in a certain domain". The generalised FBoW produces vectors of u elements, where $u = |U|$.

Given the fuzzy embeddings of the words in a sentence S, the generalised FBoW representation of S is a vector $\hat{\boldsymbol{\mu}}_S$ whose jth element $\hat{\mu}_{S,j}$ (with $j \in [1, u] \subseteq \mathbb{N}$) can be computed as

$$\hat{\mu}_{S,j} = \max_{w_i \in S} c_{S,i} \cdot \hat{v}_{w_i,j} \tag{6}$$

where $c_{S,i}$ and $\hat{v}_{w_i,j}$ are, respectively, the number of occurrences of word w_i in sentence S and the jth element of the $\hat{\mathbf{v}}_{w_i}$ vector.

The universe set can be defined in different ways, and the same applies for the universe matrix [42]. Among the possible solutions, the DynaMax algorithm for fuzzy sentence embeddings builds the universe matrix from the word embedding matrix, stacking solely the embedding vectors of the words appearing in the sentences to be compared.

Notice that in this way the resulting universe matrix is not unique, and as a consequence, neither are the embeddings. This condition can be noticed from the description of the algorithm and from the definition of the universe matrix: when comparing two sentences S_a and S_b, the universe set U used in their comparison is $U \equiv S_a \cup S_b$, so the resulting sentence embeddings have size $u = |U| = |S_a \cup S_b|$. In fact, the universe matrix is given by

$$\mathbf{U} = \left[\mathbf{u}_{w_i} \forall w_i \in U\right]^\top. \tag{7}$$

This characteristic is unfortunate as, for example, in IR, it requires a complete re-encoding of the entire document achieved for each query.

The real improvement of DynaMax is in the introduction of the fuzzy Jaccard index to compute the semantic similarity between two sentences S_a and S_b, rather than the generalisation of the FBoW, which replaced the original use of the cosine similarity [41]:

$$\hat{\sigma}_{Jaccard}\left(\hat{\boldsymbol{\mu}}_{S_a}, \hat{\boldsymbol{\mu}}_{S_b}\right) = \frac{\sum_{i=1}^{u} \min\left(\hat{\mu}_{S_a,i}, \hat{\mu}_{S_b,i}\right)}{\sum_{i=1}^{u} \max\left(\hat{\mu}_{S_a,i}, \hat{\mu}_{S_b,i}\right)}. \tag{8}$$

3 Static Fuzzy Bag-of-Words Model

Starting from the DynaMax, which evolved from the FBoW model, we developed our follow-up aimed at providing a unique matrix $\mathbf{U}$ and thus embeddings with a fixed dimension. In Fig. 1 is represented the visualisation of our approach.

3.1 Word Embeddings

Word embeddings play a central role in our algorithm as they also provide the starting point of the construction of the universe matrix. For this work, we leveraged pre-trained shallow models (more details in Sect. 4.1) for two main reasons:

- The model is encoded in a matrix where each row corresponds to a word.
- We want to provide a sentence embedding approach that does not require training, easing its accessibility.

The vocabulary of these models, composed starting from all the tokens in the training corpora, is usually more extensive than the English vocabulary, as it contains named entities, incorrectly spelt words, non-existing words, URLs, email addresses and similar. To reduce the computational effort needed to construct and use the universe matrix, we have considered some subsets of the employed word embedding model's vocabulary.

Depending on the experiment, we work with either the 100,000 most frequently used terms, the 50,000 most frequently used terms (term frequencies are given by the corpora used to train the word embedding model) or the subset composed of all the spell-checked terms present in a reference English dictionary (obtained through the Aspell English spell-checker[1]).

In the following sections, the $\check{\mathbf{W}}$ symbol refers to these as *reduced* word embedding matrices/models.

3.2 Universe Matrix

During the experiments, we tried four main approaches to build the universe matrix $\mathbf{U}$: the first two – proposed, but not explored, by the original authors of DynaMax [42] – consist, respectively, in the usage of a clustered embedding matrix and an identity matrix with the rank equal to the size of the word embeddings. Instead,

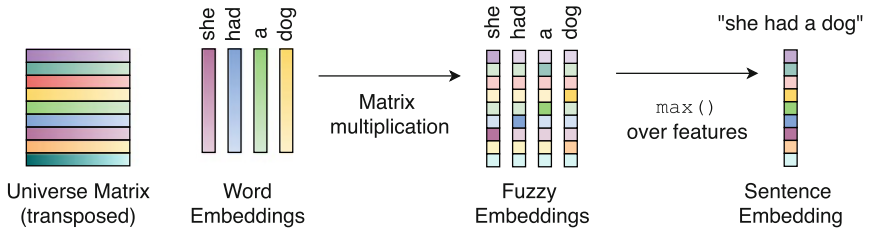

Fig. 1 Visualisation of the sentence embedding computation process using SFBoW

[1] http://aspell.net.

the third approach consists of applying multivariate analysis techniques to the word embedding matrix to build the universe one. The last approach considers the norm of the word vectors to filter out less significant words for the representation.

In the following formulae, we refer to d as the dimensionality of the word embedding vectors, while the SFBoW embedding of the singleton of word w_i is represented as $\check{\mathbf{v}}_{w_i}$. Clustering and multivariate analysis can be applied to the whole embedding vocabulary or the subsets of the vocabulary introduced in Sect. 3.1. Apart from reducing the computational time, we did so to see if these subsets are sufficient to provide a helpful representation.

3.2.1 Clustering

The idea is to group the embedding vectors into clusters and use their centroids; in this way, the fuzzy membership will be computed over the clusters—which are expected to host semantically similar words—instead of all the word singletons. The universe set is thus built out of abstract entities only, which are the centroids. Considering k centroids the k-dimensional embedding $\check{\mathbf{v}}_{w_i}$ of the singleton of word w_i is

$$\check{\mathbf{v}}_{w_i} = \mathbf{K}^\top \cdot \mathbf{u}_{w_i} = \left[\mathbf{k}_1, \ldots, \mathbf{k}_k\right]^\top \cdot \mathbf{u}_{w_i} = \mathbf{K}^\top \cdot \mathbf{W}^\top \cdot \mathbf{v}_{w_i} \tag{9}$$

where $\mathbf{k}_j$, the jth (with $j \in [1, k] \subseteq \mathbb{N}$) column of $\mathbf{K}$, corresponds to the centroid of the jth cluster. This approach generates k-dimensional word and sentence embeddings.

3.2.2 Identity

Alternatively, instead of looking for a group of semantically similar words that may form a significant group, useful for semantic similarity, we consider the possibility of re-using the word embedding dimensions (features) to represent the semantic content of a sentence. So, we just use the identity matrix as the universe, $\mathbf{U} = \mathbf{I} \in \mathbb{R}^{d \times d}$, so that $\check{\mathbf{v}}_{w_i} \in \mathbb{R}^d$ is

$$\check{\mathbf{v}}_{w_i} = \mathbf{I} \cdot \mathbf{u}_{w_i} = \mathbf{I} \cdot \mathbf{W}^\top \cdot \mathbf{v}_{w_i} \tag{10}$$

where this approach generates d-dimensional word embeddings and sentence embeddings.

3.2.3 Multivariate Analysis

The same idea moves our multivariate analysis proposal. Judging by previous results, word embeddings aggregated correctly might be sufficient to provide a semantically valid representation of a sentence.

What can bring better results might be as simple as roto-translate the reference system of the embedding representation. In this sense, we propose to use to compute the fuzzy membership, and hence the fuzzy Jaccard similarity index, over these dimensions resulting from roto-translation, expecting that this "new perspective" will expose better the semantic content. So, defining $\mathbf{U} = \mathbf{M}$, where $\mathbf{M} \in \mathbb{R}^{d \times d}$ is the transformation matrix, we have that $\check{\mathbf{v}}_{w_i} \in \mathbb{R}^d$ is

$$\check{\mathbf{v}}_{w_i} = \mathbf{M} \cdot \mathbf{u}_{w_i} = \mathbf{M} \cdot \mathbf{W}^\top \cdot \mathbf{v}_{w_i} \tag{11}$$

thus yielding d-dimensional word and sentence embeddings.

3.2.4 Vector Significance

Early analysis of shallow word embedding models showed that word vectors providing stronger semantic representation have a higher norm [34]. Moreover, when comparing the norm of the vectors with their term frequency within the training corpus, it is possible to notice that highly frequent terms, as well as rare one, have considerably smaller norm.

This concept is not anew. In fact, in the *term frequency-inverse document frequency* (TF-IDF) approach for document representation, rare words, as well as highly frequent words, should give little if any contribution to the meaning representation [7, 20]. For similar reasons, in data mining and retrieval settings, *stop words*, which are the highly frequent words in a corpus, are discarded from the document analysis.

We propose to leverage the word embeddings with a significance level above a certain (custom) threshold to build the universe matrix, to retain only the most relevant vectors. Defining $\mathbf{U} = \mathbf{L}^\top$, where $\mathbf{L} \in \mathbb{R}^{d \times d}$ is the matrix whose columns are the first n word vectors in decreasing Euclidean norm $\|\mathbf{u}_{w_i}\|_2$ order, we have that $\check{\mathbf{v}}_{w_i} \in \mathbb{R}^d$ is

$$\check{\mathbf{v}}_{w_i} = \mathbf{L}^\top \cdot \mathbf{u}_{w_i} = \left[\ldots, \mathbf{u}_{w_j}, \ldots\right]^\top \cdot \mathbf{u}_{w_i} = \mathbf{L}^\top \cdot \mathbf{W}^\top \cdot \mathbf{v}_{w_i} \tag{12}$$

where the resulting sentence embeddings have as many dimensions as the number n of retained word vectors.

4 Experiments

In order to find the best solution in terms of word embedding matrix and universe matrix, we explored various possibilities. Then, to measure the goodness of our sentence embeddings, we leveraged a series of STS tasks and compared the results with the preceding models.

4.1 Word Embeddings

For what concerns the word embeddings, we have decided to work with a selection of four models:

- Word2Vec, with 300-dimensional embeddings
- GloVe, with 300-dimensional embeddings
- fastText, with 300-dimensional embeddings
- Sent2Vec, with 700-dimensional embeddings

As shown by the word embedding models list, we are also employing a Sent2Vec sentence embedding model. The embedding matrix of this model can be used for word embeddings too. During the experiments, we focused on the universe matrix construction. For this reason, we relied on pre-trained models for word embeddings, available on the web.

4.2 Universe Matrices

The universe matrices we considered are divided into four buckets, as described in Sect. 3.2.

4.2.1 Clustering

Universe matrices built using clustering leverage four different algorithms: k-means, spherical k-means, DBSCAN and HDBSCAN.

We selected k-means and spherical k-means because they usually lead to good results; the latter was specifically designed for textual purposes, with low demand in time and computational resources. For all algorithms, we considered the same values for k (the number of centroids), which were 100, 1000, 10,000 and 25,000.

For all the values of k, we performed clustering on different subsets of the vocabulary: k-means was applied on the whole English vocabulary as well as to the top 100,000 frequently used words subset, while spherical k-means was applied

to the subset of the first 50,000 frequently used words (to reduce computational time).

We also explored density-based algorithms (DBSCAN and HDBSCAN), which do not require defining in advance the number of clusters, using Euclidean and cosine distance between the word embedding.

For what concerns DBSCAN with Euclidean distance, we varied the radius of the neighbourhood ε between 3 and 8 and worked over the same two subsets considered for k-means, while the cosine distance ε was between 0.1 and 0.55, and it was applied over the subset of the first 50, 000 frequently used words (for computational reasons, as we did for spherical k-means). Concerning HDBSCAN, we varied the smallest size grouping of clusters in the set $\{2, 4, 30, 50, 100\}$ and the minimum neighbourhood size of core samples in the set $\{1, 2, 5, 10, 50\}$. We considered this latter density-based algorithm since basic DBSCAN happens to fail with high-dimensional data.

4.2.2 Identity

This approach consists of using the identity matrix as the universe, and in this way, the singletons we use to compute the fuzzy membership are the dimensions of the word embeddings, which corresponds to the learnt features. This is the most lightweight method as it just requires to compute the word embeddings of a sentence and then the fuzzy membership over the exact d dimensions.

4.2.3 Multivariate Analysis

We adopted the principal component analysis (PCA) to get a rotation matrix to serve as a universe matrix to the SFBoW. In fact, through PCA, the d-dimensional word embedding vectors are decomposed along with the d orthogonal directions of their variance. These components are then reordered to decrease explained variance and represent our fuzzy semantic sets.

The principal component of the reduced word embedding matrix $\check{\mathbf{W}}$ is described by the matrix $\mathbf{T} = \mathbf{P}^\top \cdot \check{\mathbf{W}}$, where $\mathbf{P}$ is a $d \times d$ matrix whose columns are the eigenvectors of the matrix $\check{\mathbf{W}}^\top \cdot \check{\mathbf{W}}$. With our approach, the matrix $\mathbf{P}^\top$, sometimes called the *whitening* or *sphering transformation matrix*, serves as universe matrix $\mathbf{U}$. In this way, the SFBoW embedding of a word singleton becomes

$$\check{\mathbf{v}}_{w_i} = \mathbf{P}^\top \cdot \mathbf{u}_{w_i} = \mathbf{P}^\top \cdot \check{\mathbf{W}}^\top \cdot \mathbf{v}_{w_i} \tag{13}$$

where, as for the clustering approach, we experimented with both the whole vocabulary and the most 100,000 used words.

4.2.4 Vector Significance

As premised, we considered word embeddings norm to identify the significance of a term. We composed the universe matrix sorting the word vectors in decreasing Euclidean norm order and taking the first n. During the experiments, we varied n in the set {100, 1000, 10, 000, 25, 000}.

4.3 Data

We evaluated our SFBoW through a series of reference benchmarks; we selected the STS benchmark series, one of the tasks of the International Workshop on Semantic Evaluation (SemEval).[2]

SemEval is a series of evaluations on computational semantics; among these, the *semantic textual similarity* (STS) benchmark[3] [13] has become a reference for scoring of sentence embedding algorithms. All the previous models we are considering for comparison have been benched against STS; this is because the benchmark highlights a model capability to provide a meaningful semantic representation by scoring the correlation between model's and human's judgements. For this reason, and also to allow comparisons, we decided to evaluate SFBoW on STS.

We worked only on the English language, using the editions of STS from 2012 to 2016 [1–5]. Each year, a collection of corpora coming from different sources has been created and manually labelled; Table 1 shows a reference, in terms of support, for each edition. Thanks to the high number of samples, we are confident about the robustness of our results.

To preprocess the input text strings, we lowercased each character and tokenised in correspondence of spaces and punctuation symbols. Then, from the resulting sequence, we retained only the tokens for which a corresponding embedding was found in the vocabulary known by the model. Finally, we calculated the SFBoW sentence embedding from the word embeddings of such tokens.

The samples constituting the corpora are a pair of sentences with a human-given similarity score (the *gold labels*). The provided score is a real-valued index obtained averaging those of multiple crowd-sourced workers and is scaled in a $[0, 1] \in \mathbb{R}$

Table 1 Support of the corpora of the STS benchmark series

STS edition	2012	2013	2014	2015	2016
No. of sentence pairs	5250	2250	3750	3000	1186

[2] https://aclweb.org/aclwiki/SemEval_Portal.

[3] https://ixa2.si.ehu.eus/stswiki/index.php/Main_Page.

interval. The final goal of our work is to provide a model able to provide a score as close as possible to that of humans.

4.4 Evaluation Approach

To assess the quality of our model, we used it to compute the similarity score between the sentence pairs provided by the five tasks, and we compared the output with the target labels. The results are computed as the correlation between the similarity score produced by SFBoW and the human one, using Spearman's ρ measure [32]. SFBoW employs fuzzy Jaccard similarity index [42] to compute word similarity.

To have terms of comparison, we establish a baseline through the most straightforward models possible, the average word embedding in a sentence, leveraging three different word embedding models: Word2Vec, GloVe and fastText. We also provide results from more complex models: SIF weighting (applied to GloVe), Sent2Vec, DynaMax (built using Word2Vec, GloVe and fastText) and Sentence-BERT.

All the embedding models except DynaMax and the baselines are scored using cosine similarity; DynaMax scores are obtained using fuzzy Jaccard similarity index.

5 Results

To analyse the results of the considered reference embeddings and the approaches to build the universe matrix, we reported, respectively, the aggregated Spearman's ρ correlation in the STS benchmark in Tables 2 and 3. Through these two tables, we highlight how the choice of an embedding model rather than a universe matrix approach affected the overall SFBoW performances in the STS benchmark. Additionally, we report a comparison in terms of Spearman's ρ correlation in the STS benchmark of our SFBoW against other sentence embedding models in Table 4. The comparison values, reported in the last three rows of Table 4, belong to the SFBoW configurations that achieved the best score, among the variants we considered for the experiments, in at least one task.

5.1 Individual SFBoW Results

As reported in Table 4, fastText yields the best absolute results among the four-word embedding models, confirming the results of DynaMax. The best scores in terms of universe matrix are achieved either with identity matrix or with PCA rotation matrix,

Table 2 SFBoW aggregated results over the STS benchmark. Results are aggregated on the employed word embedding model. Total scores are weighted averages across the STS editions and are expressed as *avg.*±*std.* Bold and underlined values represent, respectively, the first and second best results of a column

Reference embedding model	Results (Spearman's ρ)					
	STS					Total
	2012	2013	2014	2015	2016	
Word2Vec	51.25 ± 4.79	42.98 ± 5.27	57.62 ± 5.92	62.74 ± 6.90	**62.81** ± 5.91	54.71 ± 5.63
GloVe	52.71 ± 5.14	43.40 ± 5.36	54.47 ± 7.20	61.55 ± 7.47	62.61 ± 6.32	54.26 ± 6.21
fastText	**54.00** ± 4.90	**44.16** ± 4.86	54.89 ± 7.60	61.62 ± 7.57	62.13 ± 7.31	54.88 ± 6.26
Sent2Vec	53.13 ± 1.46	41.48 ± 2.42	**59.17** ± 2.70	**64.81** ± 2.97	**62.81** ± 2.18	**55.91** ± 2.25

Table 3 SFBoW aggregated results over the STS benchmark. Results are aggregated on the universe matrix building approach. Total scores are weighted averages across the STS editions and are expressed as *avg.*±*std.* Bold and underlined values represent, respectively, the first and second best results of a column

Universe matrix approach	Results (Spearman's ρ)					
	STS					Total
	2012	2013	2014	2015	2016	
Clustering	53.04 ± 3.60	42.69 ± 4.17	56.42 ± 5.45	62.81 ± 5.57	62.62 ± 4.42	54.99 ± 4.58
Identity	56.90 ± 3.87	**49.45** ± 3.61	**64.56** ± 2.20	**70.59** ± 1.82	69.33 ± 4.20	**61.29** ± 3.05
Multivariate analysis	**57.53** ± 3.27	48.64 ± 3.44	64.26 ± 1.77	70.20 ± 1.89	**69.80** ± 4.02	61.27 ± 2.72
Vector significance	48.49 ± 3.53	39.61 ± 2.48	51.05 ± 5.29	56.51 ± 5.36	57.18 ± 4.53	50.04 ± 4.24

highlighting how the features yield by word embeddings provide a better semantic content representation of sentences.

To have a better understanding of the results and the performances of different universe matrices, we broke down the results along two axes. On one side, we aggregated the results distinguishing among the different embedding models (see Table 2), and on the other, we distinguished among the different approaches to build the universe matrix (see Table 3).

From Table 2, we noticed that, despite being fastText the word embedding model yielding the best performances, Sent2Vec achieved the best results on average. While the remaining models achieved on average very similar scores—all differences in Spearman's ρ are < 1—Sent2Vec detached from fastText (the second best model on average) with a difference > 1 in Spearman's ρ score. We hypothesise that this is due to the fact that Sent2Vec, different from the other embeddings, is actually a parametric sentence embedding model, which yields embeddings for single words. However, despite being different, the average results of all models are quite close, especially if compared with the differences found among average the universe matrix results.

Table 4 Comparison of results over the STS benchmark. SFBoW models are in the last block. Total scores are weighted averages across the STS editions and are expressed as *avg.*±*std.* Bold and underlined values represent, respectively, the first and second best results of a column. Inference time refers to the time, in seconds, to carry out an evaluation on the entire STS corpus

	Results (Spearman's ρ)						
	STS						
Model	2012	2013	2014	2015	2016	Total	Analysis time [s]
Word2Vec[a]	55.46	58.23	64.05	67.97	66.28	61.21 ± 5.04	–
GloVe[a]	53.28	50.76	55.63	59.22	57.88	54.99 ± 2.80	–
fastText[a]	58.82	58.83	63.42	69.05	68.24	62.65 ± 4.20	–
SIF weighting[b]	56.04	<u>62.74</u>	64.29	69.89	70.71	62.84 ± 5.54	–
Sent2Vec	56.26	57.02	65.82	74.46	69.01	63.21 ± 7.13	–
DynaMax[c]	55.95	60.17	65.32	73.93	71.46	63.53 ± 6.92	–
DynaMax[b]	57.62	55.18	63.56	70.40	71.36	62.25 ± 5.85	–
DynaMax[d]	61.32	61.71	66.87	<u>76.51</u>	<u>74.71</u>	<u>66.71</u> ± 6.10	–
Sentence-BERT	**72.27**	**78.46**	**74.90**	**80.99**	**76.25**	**75.81** ± 3.27	218.3
SFBoW[d,e,f]	61.31	51.21	<u>67.47</u>	72.90	73.88	64.55 ± 7.20	56.5
SFBoW[d,g,h]	<u>61.42</u>	51.36	66.44	72.74	73.72	64.32 ± 7.00	56.8
SFBoW[d,g,i]	60.03	51.96	66.36	72.39	73.25	63.81 ± 6.93	56.6

[a] Used as baseline
[b] Built upon a GloVe model for word embeddings
[c] Built upon a Word2Vec model for word embeddings
[d] Built upon a fastText model for word embeddings
[e] Best average score
[f] Universe matrix is the identity matrix
[g] Universe matrix is the PCA projection matrix
[h] Universe matrix is built from the English vocabulary
[i] Universe matrix is built from the top 100,000 most frequent words

From Table 3, instead, we noticed that there is a clear difference in performances among the considered approaches. Identity matrix and PCA universe matrices consistently outperform all the other considered approaches, achieving also very close scores between them—the difference between their average Spearman's ρ is only 0.02. Moreover, identity and PCA achieve scores very similar to the SFBoW predecessor (see Table 4). We hypothesise that this is due to the fact that these two techniques preserve the features extracted by the embedding models, which are very robust, as observed by other non-parametric sentence embedding models like SIF weighting.

Clustering, instead, presents way worse performances: the drop in Spearman's ρ is > 5 with respect to identity and PCA. Nevertheless, clustering scores are in line with the single word embedding model's averages.

Vector significance turned out to provide the worst overall results. We hypothesise it is due to the fact that the significance is not strongly related to the semantic representative capabilities.

5.2 Comparison with Other Models

As premised, we compare our results with three baseline models and other sentence embedding approaches, all reported in Table 4. The first group of scores is from the baselines, the second one is from other sentence embedding models, and, finally, the last group is from our SFBoW model. Additionally, the best values in each column are highlighted in bold, while the second ones are underlined.

The key features about our model, which can be derived from the results, are the following:

- Low number of parameters
- Faster inference time
- No training phase
- Results (in terms of ρ) comparable to similar models
- Fixed-size and easily re-usable embeddings

About the number of parameters, we can notice that even if Sentence-BERT outperforms all the other models in every task, it relies on a much deeper feature extraction model and was trained on a much bigger corpus. Moreover, this model requires a considerably higher computational effort without an equally consistent difference in performances. BERT alone requires more than 100 million parameters just for its base version (and above 300 million for the large one), hence taking a lot of (memory) space, not to mention the amount of time necessary for the self-supervised training and the fine-tuning. On the other hand, non-parametric models (like SIF, DynaMax or SFBoW) or shallow parametric ones (Sent2Vec) require fewer parameters: just those for the embedding matrix $|\mathcal{V}| \times d$.

A similar discourse applies to inference speed. Even though Sentence-BERT achieves the best results on all tasks, SFBoW turns out to be four times faster at inferring the similarity, as can be noticed by the reported analysis times.

Being a non-parametric model, SFBoW does not require a training phase. It may require clustering the embeddings to build the universe matrix, but our experiments showed that clustering does not yield good results. Because of its simplicity, SFBoW can generally be easily deployed, requiring only the word embedding model to compute the sentence representation. Notice also that the SFBoW algorithm is agnostic to the word embedding model.

Regarding the results we obtained, compared to other models, SFBoW provided interesting figures: either considering the majority of tasks with higher Spearman's ρ rank or higher average score, it outperforms all the baselines, as well as SIF weighting and Sent2Vec. Finally, we see as our model performs closely to its predecessor, especially considering the weighted average of the results of the single tasks. SFBoW bests out DynaMax in STS 2014 and gets almost the same results in STS 2012 (the difference is 0.01), which are the first two corpora in terms of samples; however, the difference in STS 2013 goes in favour of DynaMax.

About the comparison against DynaMax, it is worth underlining a few additional points. Firstly, in both cases, fuzzy Jaccard similarity correlates better with human

judgement as a measure of sentence similarity. Secondly, both models manage to achieve better results when using fastText word embedding, possibly underling that they lend better than other models at sentence-level combination; the baseline performances also show this.

Finally, we remind that SFBoW generates embeddings with a fixed size, resulting in much easier applicability with respect to DynaMax.

6 Conclusion

In this paper, we presented and evaluated the SFBoW model for sentence embedding. This model leverages the approaches proposed by the FBoW and DynaMax models, to compute static embeddings (in the sense of fixed-size embeddings). To extract such static embeddings, we rely on a static universe matrix. This matrix can be constructed in many different ways; thus, we explored them in order to find the most suitable. We considered approaches based on clustering, identity, multivariate analysis and vector significance. To evaluate the possible approaches, we benchmarked the model on the STS benchmark.

We divided the evaluation into an individual one, to observe the different results of the considered embeddings and approaches for the SFBoW universe matrix, and a compared one, to observe the results of SFBoW with respect to those of other sentence embedding models.

From the individual analysis, we derived that fastText and Sent2Vec are the two most suitable embeddings for our model and that identity and PCA are the most suitable universe matrix building approaches. From the compared evaluation, we derived that even if SFBoW does not outperform state-of-the-art models on STS, it performs comparably to DynaMax, its predecessor, and, different from DynaMax, yields re-usable embeddings, because of their fixed dimensionality. Due to its low computation demand (especially if compared with state-of-the-art Sentence-BERT) and re-usability of embeddings, SFBoW can be seen as a reasonable solution, especially for scenarios where low computational capabilities are essential.

In the future, we plan to carry out a deeper analysis of the results to identify the reasons behind the different scores achieved by the universe matrix approaches. Another idea for future evolution we considered is to combine the approaches we analysed to build the universe matrix, in order to extract a more robust one. For example, it would be possible to cluster the vectors with a significance above a certain threshold to obtain, possibly, better results.

Acknowledgments This work was partially supported by the European Union's Horizon 2020 project *WorkingAge* (grant agreement no. 826232).

References

1. Agirre, E., Cer, D.M., Diab, M.T., Gonzalez-Agirre, A.: Semeval-2012 task 6: A pilot on semantic textual similarity. In: Agirre, E., Bos, J., Diab, M.T. (eds.) Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012, pp. 385–393. The Association for Computer Linguistics (2012). https://www.aclweb.org/anthology/S12-1051/
2. Agirre, E., Cer, D.M., Diab, M.T., Gonzalez-Agirre, A., Guo, W.: *sem 2013 shared task: Semantic textual similarity. In: Diab, M.T., Baldwin, T., Baroni, M. (eds.) Proceedings of the Second Joint Conference on Lexical and Computational Semantics, *SEM 2013, June 13-14, 2013, Atlanta, Georgia, USA, pp. 32–43. Association for Computational Linguistics (2013). https://www.aclweb.org/anthology/S13-1004/
3. Agirre, E., Banea, C., Cardie, C., Cer, D.M., Diab, M.T., Gonzalez-Agirre, A., Guo, W., Mihalcea, R., Rigau, G., Wiebe, J.: Semeval-2014 task 10: Multilingual semantic textual similarity. In: Nakov, P., Zesch, T. (eds.) Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014, pp. 81–91. The Association for Computer Linguistics (2014). https://doi.org/10.3115/v1/s14-2010
4. Agirre, E., Banea, C., Cardie, C., Cer, D.M., Diab, M.T., Gonzalez-Agirre, A., Guo, W., Lopez-Gazpio, I., Maritxalar, M., Mihalcea, R., Rigau, G., Uria, L., Wiebe, J.: Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In: Cer, D.M., Jurgens, D., Nakov, P., Zesch, T. (eds.) Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015, pp. 252–263. The Association for Computer Linguistics (2015). https://doi.org/10.18653/v1/s15-2045
5. Agirre, E., Banea, C., Cer, D.M., Diab, M.T., Gonzalez-Agirre, A., Mihalcea, R., Rigau, G., Wiebe, J.: Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In: Bethard, S., Cer, D.M., Carpuat, M., Jurgens, D., Nakov, P., Zesch, T. (eds.) Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016, pp. 497–511. The Association for Computer Linguistics (2016). https://doi.org/10.18653/v1/s16-1081
6. Arora, S., Liang, Y., Ma, T.: 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 (2017). https://openreview.net/forum?id=SyK00v5xx
7. Baeza-Yates, R., Ribeiro-Neto, B.A.: Modern Information Retrieval - The Concepts and Technology Behind Search, Second edition. Pearson Education Ltd., Harlow, England (2011). http://www.mir2ed.org/
8. Baroni, M., Dinu, G., Kruszewski, G.: Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 238–247. Association for Computational Linguistics, Baltimore, Maryland (2014). https://doi.org/10.3115/v1/P14-1023. https://aclanthology.org/P14-1023
9. Bengio, Y., Courville, A.C., Vincent, P.: Representation learning: A review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013). https://doi.org/10.1109/TPAMI.2013.50
10. Bengio, Y., LeCun, Y., Hinton, G.E.: Deep learning for AI. Commun. ACM **64**(7), 58–65 (2021). https://doi.org/10.1145/3448250
11. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Trans. Assoc. Comput. Linguist. **5**, 135–146 (2017). https://transacl.org/ojs/index.php/tacl/article/view/999
12. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler,

E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6–12, 2020, virtual (2020). https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html
13. Cer, D.M., Diab, M.T., Agirre, E., Lopez-Gazpio, I., Specia, L.: Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In: Bethard, S., Carpuat, M., Apidianaki, M., Mohammad, S.M., Cer, D.M., Jurgens, D. (eds.) Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3–4, 2017, pp. 1–14. Association for Computational Linguistics (2017)
14. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011). http://dl.acm.org/citation.cfm?id=2078186
15. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) 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), pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/n19-1423
16. Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R.S., Urtasun, R., Torralba, A., Fidler, S.: Skip-thought vectors. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7–12, 2015, Montreal, Quebec, Canada, pp. 3294–3302 (2015). https://proceedings.neurips.cc/paper/2015/hash/f442d33fa06832082290ad8544a8da27-Abstract.html
17. LeCun, Y., Bengio, Y., Hinton, G.E.: Deep learning. Nature **521**(7553), 436–444 (2015). https://doi.org/10.1038/nature14539
18. Liu, Q., Kusner, M.J., Blunsom, P.: A survey on contextual embeddings. CoRR **abs/2003.07278** (2020). https://arxiv.org/abs/2003.07278
19. Liu, Z., Lin, Y., Sun, M.: Representation Learning for Natural Language Processing. Springer (2020). https://doi.org/10.1007/978-981-15-5573-2
20. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008). https://doi.org/10.1017/CBO9780511809071. https://nlp.stanford.edu/IR-book/pdf/irbookprint.pdf
21. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: Bengio, Y., LeCun, Y. (eds.) 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2–4, 2013, Workshop Track Proceedings (2013). http://arxiv.org/abs/1301.3781
22. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q. (eds.) 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, pp. 3111–3119 (2013). https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html
23. Osgood, C.E., Suci, G.J., Tannenbaum, P.H.: The measurement of meaning. Am. J. Sociol. **63**(5), 550–551 (1958)
24. Pagliardini, M., Gupta, P., Jaggi, M.: Unsupervised learning of sentence embeddings using compositional n-gram features. In: Walker, M.A., Ji, H., Stent, A. (eds.) Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018 (Volume 1, Long Papers), pp. 528–540. Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/n18-1049

25. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) 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, pp. 1532–1543. ACL (2014). https://doi.org/10.3115/v1/d14-1162
26. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. OpenAI blog **1**(11), 12 (2018). https://openai.com/blog/language-unsupervised/
27. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019). https://openai.com/blog/better-language-models/
28. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**, 140:1–140:67 (2020). http://jmlr.org/papers/v21/20-074.html
29. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100, 000+ questions for machine comprehension of text. In: Su, J., Carreras, X., Duh, K. (eds.) Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1–4, 2016, pp. 2383–2392. The Association for Computational Linguistics (2016). https://doi.org/10.18653/v1/d16-1264
30. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: Unanswerable questions for squad. In: Gurevych, I., Miyao, Y. (eds.) Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15–20, 2018, Volume 2: Short Papers, pp. 784–789. Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/P18-2124. https://aclanthology.org/P18-2124/
31. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) 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, Hong Kong, China, November 3–7, 2019, pp. 3980–3990. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/D19-1410
32. Reimers, N., Beyer, P., Gurevych, I.: Task-oriented intrinsic evaluation of semantic textual similarity. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 87–96. The COLING 2016 Organizing Committee, Osaka, Japan (2016). https://aclanthology.org/C16-1009
33. Salton, G.: The SMART Retrieval System: Experiments in Automatic Document Processing. Prentice-Hall Series in Automatic Computation. Prentice-Hall (1971)
34. Schakel, A.M.J., Wilson, B.J.: Measuring word significance using distributed representations of words. CoRR **abs/1508.02297** (2015). http://arxiv.org/abs/1508.02297
35. Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.Y., Gao, J., Dolan, B.: A neural network approach to context-sensitive generation of conversational responses. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 196–205. Association for Computational Linguistics, Denver, Colorado (2015). https://doi.org/10.3115/v1/N15-1020. https://aclanthology.org/N15-1020
36. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8–13, 2014, Montreal, Quebec, Canada, pp. 3104–3112 (2014). https://proceedings.neurips.cc/paper/2014/hash/a14ac55a4f27472c5d894ec1c3c743d2-Abstract.html
37. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017,

December 4–9, 2017, Long Beach, CA, USA, pp. 5998–6008 (2017). https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
38. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: Linzen, T., Chrupala, G., Alishahi, A. (eds.) Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018, pp. 353–355. Association for Computational Linguistics (2018). https://doi.org/10.18653/v1/w18-5446
39. Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: Superglue: A stickier benchmark for general-purpose language understanding systems. In: Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada, pp. 3261–3275 (2019). https://proceedings.neurips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html
40. Yih, W., He, X., Gao, J.: Deep learning and continuous representations for natural language processing. In: Mihalcea, R., Chai, J.Y., Sarkar, A. (eds.) NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31–June 5, 2015, pp. 6–8. The Association for Computational Linguistics (2015). https://doi.org/10.3115/v1/n15-4004
41. Zhao, R., Mao, K.: Fuzzy bag-of-words model for document representation. IEEE Trans. Fuzzy Syst. **26**(2), 794–804 (2018). https://doi.org/10.1109/TFUZZ.2017.2690222
42. Zhelezniak, V., Savkov, A., Shen, A., Moramarco, F., Flann, J., Hammerla, N.Y.: 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 (2019). https://openreview.net/forum?id=SkxXg2C5FX