Engineering Science and Technology, an International Journal 55 (2024) 101730

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: www.elsevier.com/locate/jestch

Full length article

Revolutionary text clustering: Investigating transfer learning capacity of SBERT models through pooling techniques

Yasin Ortakci

Department of Computer Engineering, Karabuk University, Balıklarkayası Mevkii, Merkez, 78050, Karabuk, Turkiye

ARTICLE INFO

Keywords:
SBERT
Large language models
Sentence embeddings
Text clustering
Pooling techniques

ABSTRACT

Large Language Models (LLMs), one of the most advanced representatives of neural networks, have revolutionized the field of natural language processing. Among the many applications of these models, text clustering is gaining increasing interest. In particular, the fact that LLMs digitize text more semantically and contextually than existing methods in the literature has led LLMs to produce more successful results with clustering algorithms. However, since these models are not specifically designed for text clustering, they can lead to processing times that exceed acceptable runtime thresholds. To address this challenge, the Sentence-BERT (SBERT) model has been proposed as a solution, offering the ability to accurately measure text similarity by transforming entire texts into dense, fixed-size vectors. SBERT has been integrated into various LLMs, resulting in the creation of diverse SBERT model variants. This study aims to assess the transfer learning capabilities of SBERT models in the context of text clustering. Furthermore, it investigates the influence of CLS (classification token), mean, and max pooling techniques on the performance of these models. In this direction, we applied these pooling techniques to DistilBERT, DistilRoBERTa, ALBERT, and MPNET based SBERT models and compared their performance on different corpora. The results show that there is no clear superiority among the SBERT models. However, the mean pooling emerged as the most effective method in 13 out of 16 text clustering tasks. This finding underscores the high compatibility of the mean pooling technique with SBERT models.

1. Introduction

Large Language Models (LLMs) have recently attracted a great deal of attention with the remarkable performance of advanced chatbots and have come to dominate the field of Natural Language Processing (NLP). These models [1–5] deliver state-of-the-art results that outperform existing methods by processing text in a contextualized approach, particularly in common NLP benchmark tasks such as question answering, text entailment, sentiment analysis, and semantic text similarity. These LLMs, initially pre-trained on transformer-based neural networks leveraging extensive corpora, have become significant resources for transfer learning within NLP. They also digitize texts to generate numerical text embeddings. These embeddings present semantically enriched representations of text and can be utilized in semantic similarity tasks such as text classification and clustering [6,7].

Generative Pre-Training (GPT) models [3,8,9] among the prominent representatives of LLMs, primarily concentrate on text generation, yet the embeddings they produce are not very successful in text clustering [10]. Another breakthrough example of LLMs, Bidirectional Encoder Representations from Transformers (BERT) [1], also has a limitation in generating text embeddings. Namely, even when applied to modest datasets, it causes significantly long running times during the assessment of semantic similarity between texts due to its cross-coder architecture [11]. Meanwhile, attempts have been made to employ BERT word embeddings in text clustering; however, these efforts have yielded even less favorable outcomes than the Glove method [12]. To overcome this problem, Reimers and Gurevych proposed the Sentence-BERT (SBERT) model [11]. This innovative model, which trains a Siamese BERT network to generate fixed-size vector representations called sentence embeddings, managed to exhibit enhanced performance than its ancestor BERT on text similarity tasks. Despite these advances, the research endeavors exploring text clustering with SBERT models remain limited in the literature.

In response to this existing gap, this study aims to harness sentence embeddings derived from SBERT models for text clustering and investigate the impact of diverse pooling techniques on clustering performance. To achieve this objective, sentence embeddings were extracted from four distinct text datasets using four different SBERT-based models. The K-Means clustering algorithm was subsequently employed to cluster the generated data. Furthermore, a variety of sentence embeddings were obtained by utilizing CLS (classification

E-mail address: yasinortakci@karabuk.edu.tr.

https://doi.org/10.1016/j.jestch.2024.101730
Received 13 December 2023; Received in revised form 6 April 2024; Accepted 29 May 2024
Available online 10 June 2024

token), Mean, and Max pooling techniques for each model. The study systematically evaluated the contributions of these pooling techniques to the effectiveness of the clustering process. The SBERT models used in the study encompassed DistilBERT, DistilRoBERTa, ALBERT, and MPNET. The empirical analysis revealed that no model distinctly outperformed the others; however, when considering the various pooling techniques, the Mean pooling method exhibited superior results compared to its counterparts. The remarkable performance values achieved in this study provide compelling evidence to support the applicability of SBERT-based models as robust transfer learning tools in text clustering. The contributions of this study can be summarized as follows:

- A novel approach to text clustering is proposed, which integrates K-means with various SBERT models to generate dense and fixed-size sentence embeddings. These embeddings encapsulate the contextual understanding of the entire text, considering all words in it, and move beyond traditional methods that rely on word frequencies and co-occurrences in the text.
- Even the base SBERT models, without any fine-tuning, achieve competitive results in text clustering compared to contemporary approaches existing in the literature, underscoring their ability to generate semantically rich and contextually nuanced sentence embeddings.
- A comparative analysis of SBERT models' transfer learning capabilities with the combination of diverse pooling techniques, is presented. This analysis provides valuable insights to fellow researchers in the field, shedding light on the optimal strategies for leveraging SBERT models in text clustering tasks.

The rest of the paper is organized as follows: In the next section, we present an exploration of pertinent literature concerning the creation and subsequent clustering of vector representations of textual content. In Section 3, we describe the methodology of generating sentence embeddings via SBERT-based models and the clustering process through the application of various pooling techniques. Section 4 is devoted to a comprehensive analysis of our findings from different perspectives, deriving pivotal insights. Finally, we provide a summary of our results and suggestions for potential avenues of future research.

2. Related work

Since clustering algorithms inherently operate on numerical data, the initial step in clustering textual information involves the conversion of text into numerical representations. This conversion places each text in a separate position within a vector space. In particular, the methodologies employed for this transformation are designed to position semantically similar texts in close proximity to each other. Two common early representatives of this digitization effort are the Bag of Words (BoW) and the Term Frequency-Inverse Document Frequency (TF-IDF) methods. The BoW creates a statistical text representation by considering only the frequency of words within the text. In contrast, TF-IDF approach enriches this representation by considering the frequency of words in the entire corpus. In the literature, [13] undertakes text clustering using different clustering algorithms based on BoW, whereas [14] introduces a network-based BoW variant that integrates semantic considerations besides word frequencies. Similarly, [15,16] harness TF-IDF as a feature extraction mechanism for document clustering and categorize articles spanning various subjects into clusters.

As these two methods only focus on word repetition not considering the semantics of text, they frequently encounter challenges in capturing the meaning within textual data. In response to this limitation, neural network-driven Word2Vec strategies have been formulated [17]. These approaches improve word representations by incorporating neighboring words, thus mitigating the semantic deficiency of the aforementioned techniques. Like Word2Vec, the Glove [12] and FastText [18] techniques similarly take textual semantics into account by providing dense, fixed-size word embeddings. However, these methods also have constraints in generating word embeddings for words with more than one meaning.

LLMs, representing text both semantically and contextually more realistically than other methods, have recently been used in embedding production for text clustering purposes. Studies such as [19,20] have leveraged BERT as a tool to generate word embeddings for text clustering, while [21] employed BERT for spam email classification. [22] conducted an exhaustive evaluation of LLMs, investigating the effectiveness of different models in text clustering specifically for intent detection. One of the rare studies using SBERT for text clustering, [23] focused on categorizing scientific articles into four domains: "eaerospace", "materials science", "computer", and "life science". [24] engaged in a comparative analysis of two SBERT models based on BERT and ALBERT across various downstream NLP tasks.

On the other hand, the existing literature contains numerous short text clustering methods that address the challenges of high dimensionality and sparsity in generating numerical representations of terms. One prominent method is Non-Negative Matrix Factorization (NMF) [25] that decomposes the sparse term-document matrix into two lower-dimensional matrices: term-topic and topic-document. This approach leverages term correlations for weighting and achieves successful text clustering results. The same researchers also proposed Normalized Cut (Ncut) [26], which is derived from NMF and improved performance by incorporating term co-occurrence information instead of inverse document frequency.

Probabilistic topic modeling is a common approach in contemporary text clustering research. One such method, Biterm Topic Model (BTM) [27], extracts the co-occurrence patterns of the terms in the corpus as biterms and leverages them for text clustering. However, a notable limitation of BTM lies in its reliance on only biterm interactions between terms, neglecting higher-order word co-occurrences. Yin and Wang developed a collapsed Gibbs Sampling algorithm for Dirichlet Multinomial Mixture (GSDMM) model for short text clustering [28]. GSDMM can automatically determine the number of clusters and achieves high performance in clustering by focusing on homogeneity and completeness criteria. Similarly, Chen et al. proposed a non-parametric topic model (NPMM) [29] that incorporates pre-trained Glove word embeddings into the Dirichlet model, eliminating the need for manual hyperparameter tuning for topic generation. However, both NPMM and GSDMM can suffer from high memory consumption when processing large datasets with numerous clusters. For evolving short text streams with temporal dependencies, Kumar et al. proposed the non-parametric Dirichlet model with episodic inference (EINDM) [30], which considers contextual relationships between words. Nevertheless, EINDM is more suitable for streaming environments.

Moving beyond two-term correlation, Akritidis et al. introduced VEPCH [31], a two-stage method which utilizes the co-occurrence of three or more terms for more realistic text clustering. In the first stage, VEPCH projects text vectors to a lower dimension, identifies dominant projections for each vector, and groups documents with the same dominant projection. A subsequent refinement stage involving inter-cluster transfers and merges leads to the final clusters. The same group developed VEPH [32] as an extension of VEPCH, and it differs in several aspects: VEPH employs the tp-idf approach instead of tf-idf, integrating term positions in term weighting, creates singleton clusters during refinement, and merges them using a hierarchical approach similar to the agglomerative clustering algorithm. VEPCH and VEPH have demonstrated considerable success in short-text clustering endeavors. In addition to the aforementioned studies, [33,34] provide a comprehensive review of recent studies on text clustering.

Our study proposes a solution to the problem of sparsity and high dimensionality in text clustering by utilizing SBERT based models to generate dense sentence embedding vectors. These models generate fixed dimensional text embeddings for each document in a corpus by taking into account all the terms in a document, rather than relying solely on word co-occurrences. Thus, it produces more realistic numerical representations of text. At the same time, this study contributes to the literature by quantifying the transfer learning capabilities of

various SBERT models when applied to text clustering tasks. Moreover, it stands out by meticulously analyzing these models' synergy with different pooling techniques and indicating their effectiveness in practical applications.

3. Text clustering with SBERT

Diverging from conventional LLMs, SBERT-based models do not produce word-level embeddings, but representations of the whole text, called sentence embeddings. This distinctive approach enables the creation of semantically and contextually powerful sentence embeddings, which empowers the effective clustering of textual content. SBERT was built on the BERT network and subsequently modified for similarity tasks such as semantic search and clustering.

3.1. BERT model

In 2018, Google AI introduced BERT [35], a powerful machine learning model that produces state-of-the-art results in many NLP tasks. BERT was built upon the Transformer introduced by Vaswani et al. [36], which is a multi-layered neural network designed for processing text using a revolutionary mechanism called self-attention. This mechanism enables BERT to dynamically prioritize and comprehend the most relevant elements within the text. It also assigns varying degrees of importance to different segments of the input sequence. By doing so, BERT adeptly captures intricate word relationships, thus significantly enhancing its effectiveness in tackling complex NLP challenges. During the pre-training, BERT was fed with a massive corpus comprising Wikipedia and BooksCorpus [37], targeting two unsupervised learning objectives: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). In MLM, a percentage of input tokens are masked, and the model is trained to predict these masked tokens based on the surrounding tokens. In NSP, the model determines whether a pair of input sentences are coherent and related, assessing if they form two consecutive sentences that complement each other.

Fig. 1 shows the architecture of the base BERT model. In the initial stage, the embedding layer of the model converts the input text into tokenized vectors using the WordPiece method [38]. These vectors are combined with segment vectors (such as CLS, SEP, and padding) and positional encoding vectors that reference the positions of the tokens within the text. The resulting tokenized vectors become the input of a stack of encoders. While transformers typically have an architecture that combines encoder and decoder blocks, BERT utilizes only the encoder block due to its focus on unsupervised learning. Each encoder layer is composed of two sub-layers: Multi-Head Self-Attention and Feed-Forward Neural Network. Multi-Head Self-Attention analyzes contextual dependencies among all words in the text by computing attention scores across the sequence. Meanwhile, the Feed-Forward Neural Network enhances the model's capacity by adding non-linearity to the model in order to capture more complex relationships between words. The base model of BERT consists of 12 sequential encoder layers, each featuring 12 self-attention heads. The word embeddings produced by each encoder are the input for the subsequent encoder layer. The final encoder layer generates 768-dimensional word embeddings for each token in the text, serving as the ultimate outputs of the BERT model, known as BERT embeddings. Unlike traditional models that generate individual word embeddings, BERT constructs word embeddings based on the entire context of the text. BERT's unique capability to capture both left and right contexts during the pre-training phase sets it apart from other models. This characteristic enables BERT to generate contextually and semantically rich word embeddings. These embeddings can subsequently be fine-tuned for various downstream NLP tasks.

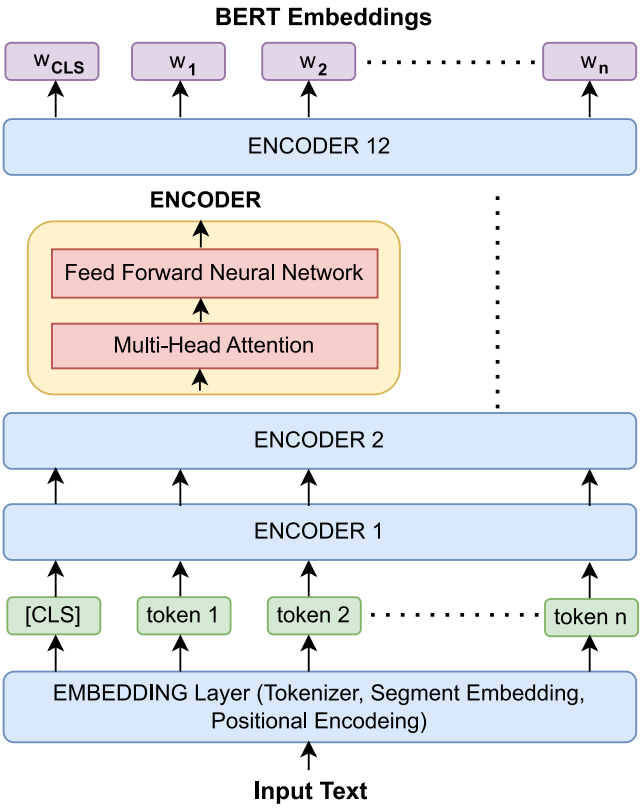

Fig. 1. The architecture of the base BERT model.

3.2. Sentence-BERT (SBERT)

Although BERT excels at understanding individual words within context, it is not specifically designed to capture the overall meaning of sentences or longer text passages effectively. This can limit its effectiveness in tasks like semantic search or finding similar text. Additionally, its capacity to accurately detect text similarities is impeded by substantial computational overhead and time. To address these limitations, Reimers and Gurevych introduced SBERT [11], which fine-tunes BERT on sentence similarity tasks. SBERT utilizes a Siamese network, which consists of two identical BERT models that share the same weights and architecture (Fig. 2). During training, this Siamese network processes sentence pairs labeled for semantic similarity, with each sentence being inputted into a separate but identical BERT model. The outputs of both BERT models which include the contextualized embeddings of each word in the input text, are combined through a pooling operation (such as averaging, max-pooling, or using the CLS token). This aggregates the word-level information into a single, fixed-size vector called sentence embeddings, representing the entire sentence's meaning. Therefore, the pooling enables SBERT to focus on the holistic representation of a sentence rather than individual word embeddings. After pooling, two separate sentence embeddings generated by the Siamese BERT models are compared with the cosine similarity, which measures how similar these two vectors are in a high-dimensional space. Subsequent to comparison, the CosineSimilarityLoss is employed as a loss function in the training. This loss function rewards the model when the distance between embeddings of dissimilar sentences is beyond a margin and penalizes it if the distance between embeddings of similar sentences is greater than the margin. The Siamese network's weights are iteratively tuned through loss function optimization across the entire training set to minimize the loss.

As a result, SBERT generates embeddings for input sentences by passing them through the trained Siamese network and the pooling

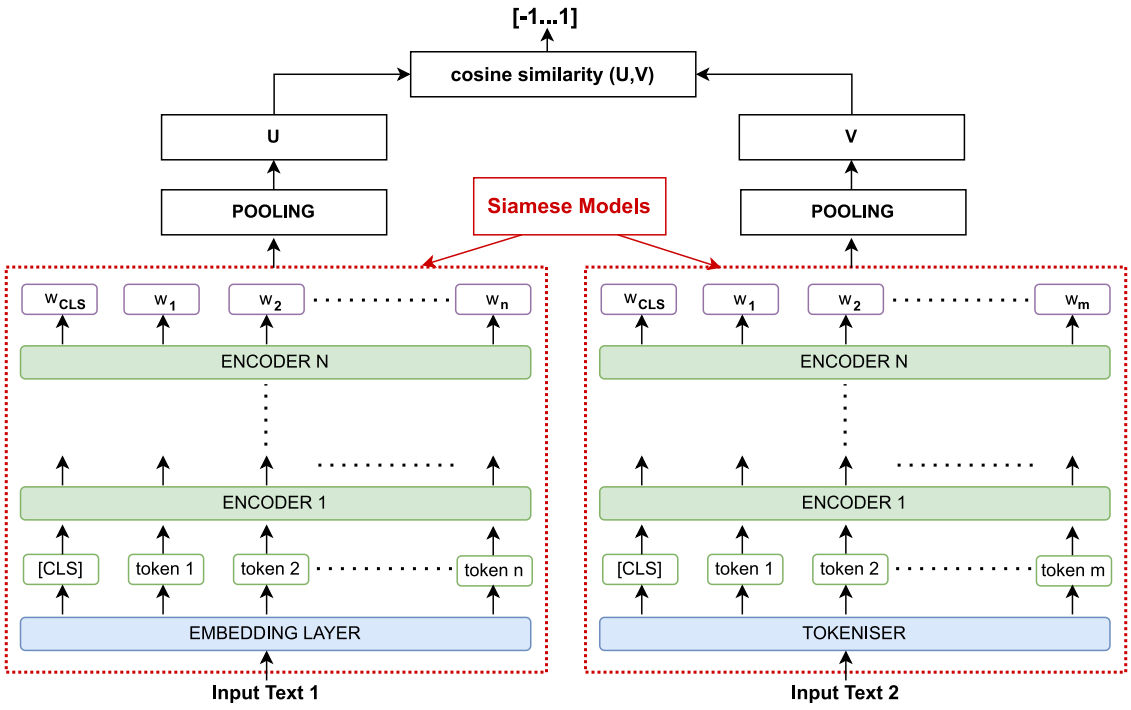

Fig. 2. The architecture of SBERT-based models.

layer. These embeddings ensure that similar sentences are embedded closer to each other in the vector space compared to dissimilar sentences. Therefore, SBERT-produced sentence embeddings offer a powerful and efficient solution for text clustering. Even though it was initially developed only for BERT and RoBERTa [39] and was not designed for transfer learning, SBERT's application domain has expanded over time. Numerous LLMs rooted in the transformer-based BERT architecture have been integrated into the SBERT framework, and different SBERT pre-trained models, termed sentence transformers, have been created. In this context, this study employs sentence transformers obtained by modifying the DistilBERT, DistilRoBERTa, ALBERT, and MPNET LLMs.

3.2.1. *DistilBERT*

DistilBERT [39] is built using knowledge distillation from a wise teacher model (BERT) and represents a smaller, faster, and more lightweight version of BERT. During distillation, DistilBERT learns from the intermediate layers' outputs of BERT, rather than directly from its final outputs. This process leverages a combination of three loss functions (Distillation Loss, Masked Language Modeling Loss, and Cosine Embedding Loss) in converting BERT into DistilBERT. Distillation reduced the number of parameters by 40%, halved the number of layers, and accelerated the model's performance by 60% compared to the teacher model. Despite these reductions, the model retains 97% of the language understanding capacity of the teacher model.

3.2.2. *DistilRoBERTa*

DistilRoBERTa [39] is a distilled variant of RoBERTa (Robustly optimized BERT approach) [2], a modification of BERT model that shares the same transformer architecture. While retaining this architecture, RoBERTa introduces various alterations during training, including using a larger dataset, discarding the NSP task, and applying a dynamic masking model as an alternative to MLM. Additionally, RoBERTa diverges from BERT by employing Byte-pair Encoding [40] for word tokenization instead of WordPiece method. During the distillation period, DistilRoBERTa utilizes the same knowledge distillation technique as implemented in DistilBERT. This allows knowledge transfer from the teacher model, RoBERTa, to DistilRaBERTa, reducing its size without compromising efficiency.

Table 1
Sentence transformer models.

Base model	Sentence transformer name
DistilBERT	multi-qa-distilbert-cos-v1
DistilRoBERTa	all-distilroberta-v1
ALBERT	paraphrase-albert-small-v2
MPNET	all-mpnet-base-v2

3.2.3. *ALBERT*

ALBERT (A Lite BERT) [41] is a compact variation of BERT designed to decrease BERT's large memory consumption and computational cost. ALBERT differs from BERT in its training methodology, marked by three key operations. Firstly, it adopts the Factorized Embedding Parameterization technique, dividing embedding matrices into smaller segments. Additionally, it includes the Sentence-Order Prediction (SOP) task as an additional objective to the MLM. ALBERT also introduces a cross-layer parameter sharing technique that enables parameter sharing across encoder layers, significantly reducing the model's parameter count. As a result, ALBERT can achieve comparable or superior performance to BERT on numerous NLP tasks while using fewer parameters.

3.2.4. *MPNET*

MPNET [42] (Multi-headed Masked Language Modeling with Permuted Language Modeling) is designed to address the limitations of BERT. Although MPNET uses a transformer architecture similar to BERT, it employs permuted language modeling that handles multiple masked tokens simultaneously, along with MLM in pre-training. Unlike BERT, MPNET does not implement the NSP task. Moreover, MPNET incorporates auxiliary location information of words to better understand word order and sentence structure. Therefore, it is an LLM specifically designed to improve performance in long-context language understanding tasks.

Table 1 lists the full names of the sentence transformers we utilized along with their base LLMs. Each model produces a 768-dimensional word embedding for every token in the input text. For instance, consider the semantic similarity comparison of two texts, one comprising

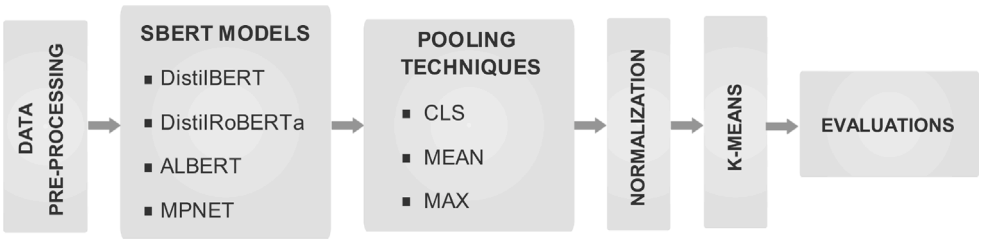

Fig. 3. The overview of text clustering with SBERT-based models.

20 tokens and the other containing 15 tokens. Each text is fed into a separate branch of the Siamese model. The first branch generates a word embedding vector with dimensions of 20 × 768, while the second branch produces a word embedding vector with dimensions of 15 × 768. However, directly comparing these different dimensional vectors poses challenges, necessitating the equalization of their sizes. To address this issue, sentence transformers incorporate the pooling techniques, which combine different dimensional word embedding vectors into a single 768-dimensional sentence embedding vector for each text, as shown in Fig. 2. Thus, the similarity of the two texts can be computed for clustering purposes. In this study, we employed three distinct pooling techniques, namely CLS, Mean, and Max.

3.3. CLS pooling

CLS is a special token added at the beginning of each text in BERT-based models. It holds the pivotal role of encapsulating the overall meaning of the text. During the training process of the models, the CLS token, like other word embeddings, undergoes refinement. The resulting CLS output, situated in the last encoder layer of the model, manifests as a representative word embedding primarily employed for classification tasks.

3.4. Mean pooling

Mean pooling creates a sentence embedding of the whole text by averaging the embeddings of all words in the text. This technique tends to be more effective when applied to longer texts, as it is able to cover all the words present in the text content.

3.5. Max pooling

Max pooling involves a systematic scan across all word embeddings within the given text, identifying the maximum values present within each dimension. The sentence embedding is then formed with these maximum values. This method highlights the key elements intrinsic to the text, constructing a representation that turns around these key components.

Fig. 3 illustrates the general methodology of the study. The initial step of this study encompassed the application of text pre-processing to the corpora. During this phase, all non-alphabetic characters were eliminated from the texts. Moreover, all URL addresses were removed from the texts. Subsequently, sentence embeddings were generated from the processed data through the utilization of three distinct pooling techniques applied to SBERT models. To ensure robustness, the sentence embeddings were then normalized using the L2-Norm [43] approach to rectify any outliers. Following the normalization, the data were clustered via the K-Means algorithm [44]. Notably, K-Means has been stated in the literature as superior to alternative clustering algorithms in critical metrics, including accuracy, scalability, and speed [45]. The final outcomes indicate the contributions of the models and pooling techniques to the overall clustering performance.

Table 2
The details of the corpora datasets.

Dataset	Number of samples	Number of clusters
Yahoo-Answer	4000	10
DBpedia	7000	14
AG News	4000	4
UCI News Aggregator	50 138	823

4. Experiments and results

This section presents a series of comprehensive analyses to examine the efficiency of SBERT models in the context of text clustering and the impact of pooling techniques on them. The experiments were conducted on an Intel Core i7-10750H 2.60 GHz CPU with 32 GB of RAM, running on Windows 11 Pro. The K-means algorithm was run 20 times, each with distinct random seeds, across various SBERT models and pooling technique combinations. The maximum number of K-means iterations was set to 300.

4.1. Datasets

Our experiments were carried out on four different datasets. The first one originates from Yahoo Answers [46], deriving from the platform's questions and answers section. This dataset consists of a spectrum of 10 distinct topics. The subset of this corpus used in the study consists of a total of 4000 texts, evenly distributed across these 10 topics, with each topic housing 400 instances. The objective is to categorize the texts into the correct topics by combining the question and answer fields in the text.

The second dataset, DBpedia [46], draws its source from the DBpedia repository, encompassing textual data spanning 14 diverse topics originating from the year 2014. Each of these topics is represented by 500 samples, resulting in a total dataset size of 7000 records.

The third dataset is the AG News corpus [46], obtained from ComeToMyHead. It comprises news articles across four distinct categories: "World", "Sports", "Business" and "Sci/Tech". The dataset contains 1000 articles for each category and cumulatively 4000 records.

The final dataset is a subset of the UCI News Aggregator dataset, which comprises news articles sourced from a web aggregator [31]. This subset includes the title of the first 50138 articles, which is grouped into 823 distinct stories (clusters). News article titles are typically short, with an average length of seven terms. A summary of the dataset including the total number of records and the number of clusters, is presented in Table 2.

4.2. Evaluation metrics

To evaluate the efficacy of the clustering outcomes, we utilized widely adopted evaluation metrics, including Normalized Mutual Information, Adjusted Rand Index, Completeness, and Homogeneity.

Normalized Mutual Information (NMI) is a prevalent metric for evaluating clustering algorithm performance. It quantifies the similarity between the clustering solution (C) and ground truth clustering (G) by

Table 3
Clustering results of Yahoo-Answer dataset.

Model	Method	NMI ± SD	ARI ± SD	HOM ± SD	COMP ± SD
DistilBERT	**CLS**	**0.434 ± 0.009**	**0.352 ± 0.013**	**0.431 ± 0.009**	**0.437 ± 0.009**
	Mean	0.337 ± 0.015	0.234 ± 0.021	0.330 ± 0.015	0.345 ± 0.014
	Max	0.262 ± 0.019	0.138 ± 0.023	0.243 ± 0.020	0.285 ± 0.018
DistilRoBERTa	CLS	0.397 ± 0.011	0.295 ± 0.021	0.393 ± 0.012	0.400 ± 0.009
	Mean	**0.404 ± 0.016**	**0.322 ± 0.017**	**0.393 ± 0.016**	**0.416 ± 0.016**
	Max	0.318 ± 0.024	0.227 ± 0.027	0.309 ± 0.023	0.327 ± 0.025
ALBERT	CLS	0.367 ± 0.007	0.308 ± 0.005	0.366 ± 0.006	0.368 ± 0.007
	Mean	**0.386 ± 0.005**	**0.325 ± 0.007**	**0.384 ± 0.005**	**0.389 ± 0.005**
	Max	0.254 ± 0.012	0.174 ± 0.013	0.245 ± 0.014	0.263 ± 0.011
MPNET	CLS	0.096 ± 0.007	0.052 ± 0.005	0.096 ± 0.007	0.096 ± 0.007
	Mean	**0.306 ± 0.031**	**0.219 ± 0.030**	**0.302 ± 0.031**	**0.310 ± 0.031**
	Max	0.252 ± 0.025	0.162 ± 0.020	0.244 ± 0.024	0.261 ± 0.027

calculating the mutual information they share, which is then normalized by the entropy in the individual clusterings [47]. NMI ranges from 0 to 1; a score of 1 denotes perfect agreement between C and G, while a score of 0 signifies no mutual information between them.

Another widely used metric, Adjusted Rand Index (ARI) [48], uses a randomness model to measure the similarity between C and G. ARI ranges from −1 to 1; a value of 1 indicates an exact match between C and G, 0 denotes no improvement over random clustering, and −1 indicates complete dissimilarity between them.

Completeness (COMP) and Homogeneity (HOM) are two key measures used to assess the quality of clustering results [49]. COMP focuses on the compactness of clusters by ensuring that all data points belonging to the same class are grouped together in a single cluster. In contrast, HOM emphasizes the purity of clusters in terms of class membership, aiming for clusters that contain only data points belonging to the same class. Both COMP and HOM range from 0 to 1, with values closer to 1 indicating a more successful clustering performance. Balancing these two measures is crucial for achieving an effective clustering solution. The formulas of NMI, ARI, COMP and HOM are given in Eq. (1), (2), (3), and (4), respectively.

$$NMI = \frac{-2\sum_i\sum_j\left(n_{ij}\log\left(\frac{N*n_{ij}}{n_i*n_j}\right)\right)}{\sum_i\left(n_i*\log\left(\frac{n_i}{N}\right)\right)+\sum_j\left(n_j*\log\left(\frac{n_j}{N}\right)\right)} \quad (1)$$

$$ARI = \left(\frac{(x+y)}{\binom{N}{2}} - \frac{\sum_i\binom{n_i}{2}*\sum_j\binom{n_j}{2}}{\binom{N}{2}}\right) \Big/ \left(1 - \frac{\sum_i\binom{n_i}{2}*\sum_j\binom{n_j}{2}}{\binom{N}{2}}\right) \quad (2)$$

$$COMP = 1 - \left(\frac{\sum_i\sum_j\left[\frac{n_{ij}}{N}*log\left(\frac{n_{ij}}{n_j}\right)\right]}{\sum_i\left[\frac{n_i}{N}*log\left(\frac{n_i}{N}\right)\right]}\right) \quad (3)$$

$$HOM = 1 - \left(\frac{\sum_j\sum_i\left[\frac{n_{ji}}{N}*\log\left(\frac{n_{ji}}{n_i}\right)\right]}{\sum_j\left[\frac{n_j}{N}*\log\left(\frac{n_j}{N}\right)\right]}\right) \quad (4)$$

Where, N is the total number of data points in the dataset. n_{ij} is the number of data points that are in both cluster i in C and cluster j in G. n_i is the number of data points in cluster i of C. n_j is the number of data points in cluster j of G. x is the number of data point pairs in the same cluster in both C and G. y is the number of data point pairs in the distinct clusters within both C and G. $\binom{N}{2}$ is the total number of possible pairs of data points and can be calculated using the formula $N(N-1)/2$.

4.3. Results

This section presents the experimental results for SBERT models and pooling techniques across all datasets and metrics, along with their standard deviations. Our analysis primarily relies on NMI since the results of all four evaluation metrics are consistently parallel. Table 3 shows the results of our proposed method on the Yahoo Answer corpus, with their respective standard deviations. Only the DistilBERT sentence transformer produced the most successful results with the CLS pooling technique across all metrics, whereas the other three sentence transformers displayed optimal results when coupled with the Mean technique. The best result came from the DistilBERT-CLS duo with an NMI value of 0.434. A comparative evaluation of the SBERT models based on their highest accomplishments reveals the following ranking of success: DistilBERT, DistilRoBERTa, ALBERT, and MPNET. To rank the pooling techniques, an average value was computed for each technique across all models. For instance, to calculate the average performance of the Mean technique on the Yahoo-Answer dataset, the metrics produced by the DistilBERT, DistilRoBERTa, ALBERT and MPNET models using the Mean technique on this dataset are averaged. The average performance of the CLS and Max techniques was similarly determined. By these measures, the Mean pooling technique emerged as the most successful on this dataset, yielding an average NMI of 0.258. It was followed by CLS with an average of 0.324, and then Max with an average NMI of 0.271.

The outputs for the DBpedia dataset are detailed in Table 4. All models, except ALBERT, reached their peak performance when paired with the Mean pooling technique. ALBERT model exhibited superior results with the CLS pooling technique. The most successful pair on this dataset is MPNET-Mean, with an NMI value of 0.855. The ranking of the models for this dataset is as follows: MPNET, DistilRoBERTa, DistilBERT, and ALBERT. Considering the individual performance of the pooling techniques, the hierarchy is as follows: Mean (0.808), Max (0.740), and CLS (0.731).

Table 5 lists the findings from the AG News dataset. Similar to the DBPedia results, ALBERT once more reached its optimum accuracy when coupled with the CLS. The other models exhibited superior performance with the Mean pooling technique. DistilBERT-Mean couple achieved the most favorable outcome on this dataset, achieving an NMI value of 0.579. When measuring the success of the models on this dataset, the order unfolds as follows: DistilBERT, DistilRoBERTa, ALBERT, and MPNET. The ranking of the pooling techniques is Mean (0.525), CLS (0.464), and Max (0.449).

Table 6 illustrates the outcomes for the UCI News Aggregator dataset. In contrast to the other datasets, all models achieved their best performance using the Mean pooling technique. Specifically, the MPNET-Mean pair attained the highest NMI score (0.854), with the model ranking order being MPNET, DistilRoBERTa, DistilBERT, and ALBERT. Regarding the pooling techniques, the ranking is as follows: Mean (0.845), Max (0.839), and CLS (0.821).

Fig. 4 demonstrates the NMI value of 48 experiments, covering four SBERT models and three pooling techniques across the four datasets. In 13 out of 16 experiments comparing the pooling techniques, the Mean method outperformed the others. In the remaining three experiments, the CLS technique yielded the most optimal results. These results highlight that Mean pooling is the most preferable technique for text clustering with SBERT models independent of model selection.

Table 4
Clustering results of DBpedia dataset.

Model	Method	NMI ± SD	ARI ± SD	HOM ± SD	COMP ± SD
DistilBERT	CLS	0.709 ± 0.013	0.544 ± 0.029	0.697 ± 0.015	0.720 ± 0.012
	Mean	**0.817 ± 0.028**	**0.700 ± 0.058**	**0.807 ± 0.031**	**0.827 ± 0.025**
	Max	0.782 ± 0.024	0.633 ± 0.039	0.763 ± 0.025	0.801 ± 0.025
DistilRoBERTa	CLS	0.790 ± 0.015	0.680 ± 0.033	0.781 ± 0.017	0.799 ± 0.013
	Mean	**0.833 ± 0.022**	**0.756 ± 0.048**	**0.825 ± 0.025**	**0.841 ± 0.019**
	Max	0.732 ± 0.015	0.584 ± 0.025	0.717 ± 0.016	0.749 ± 0.014
ALBERT	**CLS**	**0.756 ± 0.017**	**0.648 ± 0.034**	**0.747 ± 0.019**	**0.766 ± 0.016**
	Mean	0.728 ± 0.021	0.612 ± 0.037	0.722 ± 0.023	0.734 ± 0.019
	Max	0.629 ± 0.026	0.473 ± 0.032	0.615 ± 0.027	0.643 ± 0.026
MPNET	CLS	0.669 ± 0.023	0.539 ± 0.028	0.659 ± 0.024	0.679 ± 0.021
	Mean	**0.855 ± 0.018**	**0.800 ± 0.041**	**0.851 ± 0.021**	**0.859 ± 0.016**
	Max	0.816 ± 0.020	0.702 ± 0.045	0.804 ± 0.023	0.828 ± 0.017

Table 5
Clustering results of AG News dataset.

Model	Method	NMI ± SD	ARI ± SD	HOM ± SD	COMP ± SD
DistilBERT	CLS	0.487 ± 0.061	0.442 ± 0.093	0.475 ± 0.064	0.500 ± 0.059
	Mean	**0.579 ± 0.004**	**0.584 ± 0.010**	**0.576 ± 0.005**	**0.582 ± 0.003**
	Max	0.443 ± 0.048	0.387 ± 0.071	0.433 ± 0.051	0.453 ± 0.046
DistilRoBERTa	CLS	0.504 ± 0.004	0.504 ± 0.011	0.529 ± 0.006	0.552 ± 0.003
	Mean	**0.531 ± 0.003**	**0.523 ± 0.006**	**0.530 ± 0.003**	**0.558 ± 0.002**
	Max	0.492 ± 0.019	0.482 ± 0.030	0.474 ± 0.029	0.513 ± 0.018
ALBERT	**CLS**	**0.508 ± 0.004**	**0.537 ± 0.004**	**0.508 ± 0.004**	**0.509 ± 0.004**
	Mean	0.500 ± 0.006	0.512 ± 0.010	0.497 ± 0.006	0.502 ± 0.005
	Max	0.413 ± 0.009	0.370 ± 0.021	0.401 ± 0.011	0.425 ± 0.007
MPNET	CLS	0.356 ± 0.033	0.260 ± 0.060	0.334 ± 0.032	0.382 ± 0.036
	Mean	**0.490 ± 0.017**	**0.458 ± 0.026**	**0.481 ± 0.018**	**0.499 ± 0.017**
	Max	0.447 ± 0.041	0.401 ± 0.064	0.436 ± 0.043	0.457 ± 0.038

Table 6
Clustering results of UCI News Aggregator dataset.

Model	Method	NMI ± SD	ARI ± SD	HOM ± SD	COMP ± SD
DistilBERT	CLS	0.803 ± 0.002	0.355 ± 0.006	0.817 ± 0.001	0.790 ± 0.002
	Mean	**0.844 ± 0.001**	**0.414 ± 0.006**	**0.856 ± 0.001**	**0.832 ± 0.001**
	Max	0.841 ± 0.001	0.401 ± 0.006	0.850 ± 0.001	0.833 ± 0.002
DistilRoBERTa	CLS	0.843 ± 0.001	0.412 ± 0.004	0.854 ± 0.001	0.832 ± 0.001
	Mean	**0.850 ± 0.001**	**0.423 ± 0.006**	**0.861 ± 0.001**	**0.839 ± 0.001**
	Max	0.841 ± 0.001	0.403 ± 0.005	0.849 ± 0.001	0.834 ± 0.001
ALBERT	CLS	0.790 ± 0.001	0.339 ± 0.004	0.802 ± 0.001	0.778 ± 0.001
	Mean	**0.833 ± 0.001**	**0.395 ± 0.004**	**0.846 ± 0.001**	**0.821 ± 0.001**
	Max	0.823 ± 0.001	0.363 ± 0.006	0.830 ± 0.002	0.816 ± 0.002
MPNET	CLS	0.849 ± 0.001	0.418 ± 0.006	0.859 ± 0.001	0.839 ± 0.001
	Mean	**0.854 ± 0.001**	**0.431 ± 0.004**	**0.865 ± 0.001**	**0.844 ± 0.001**
	Max	0.851 ± 0.001	0.420 ± 0.005	0.860 ± 0.001	0.841 ± 0.001

4.4. Performance comparison

To demonstrate the effectiveness of SBERT models in text clustering, we conducted a comparative evaluation using recent research findings from the literature. We compared the performance of GSDMM, VEPCH, and VEPH methods, which are acknowledged for their state-of-the-art results in text clustering, against our proposed SBERT models. We used the UCI News Aggregator dataset as a benchmark and NMI as the evaluation metric. We focused on mean pooling results for comparison, as all SBERT models exhibited optimal performance with this pooling method on this dataset. Table 7 presents the comparative results, revealing that all SBERT models outperformed the GSDMM method. Notably, among these methods, MPNET+Mean Pooling was the most effective text clustering method. Additionally, our other SBERT models also demonstrated competitive results against both VEPCH and VEPH. MPNET's superior performance compared to other models may be due to the alignment of the corpora used in its pre-training with the UCI News Aggregator dataset.

Regarding the computational efficiency of the models, Table 7 displays the runtimes of each method. Since the experiments were performed on computers with identical hardware configurations, comparing the runtimes allows for an assessment of their computational efficiency. The experiments show that the SBERT-based models required more time to produce solutions compared to others. This delay is mainly due to the time needed by the SBERT models to generate sentence embeddings of the texts in the dataset. For instance, the clustering process of the MPNET+Mean Pooling pair takes a total of 925.2 s, of which 836.9 s is for embedding generation and 88.3 s is for clustering with K-Means. Similarly, the runtimes of other SBERT models

Table 7
Performance evaluation of various clustering methods.

Method	NMI	Time(sec)
GSDMM [32]	0.820	453.9
VEPCH [31]	0.851	10.2
VEPH [32]	0.853	9.4
DistilBERT+Mean Pooling	0.844	427.8 + 91.8=519.6
DistilRoBERTa+Mean Pooling	0.850	428.4 + 84.5=512.9
ALBERT+Mean Pooling	0.833	558.3 + 95.2=653.5
MPNET+Mean Pooling	**0.854**	836.9 + 88.3=925.2

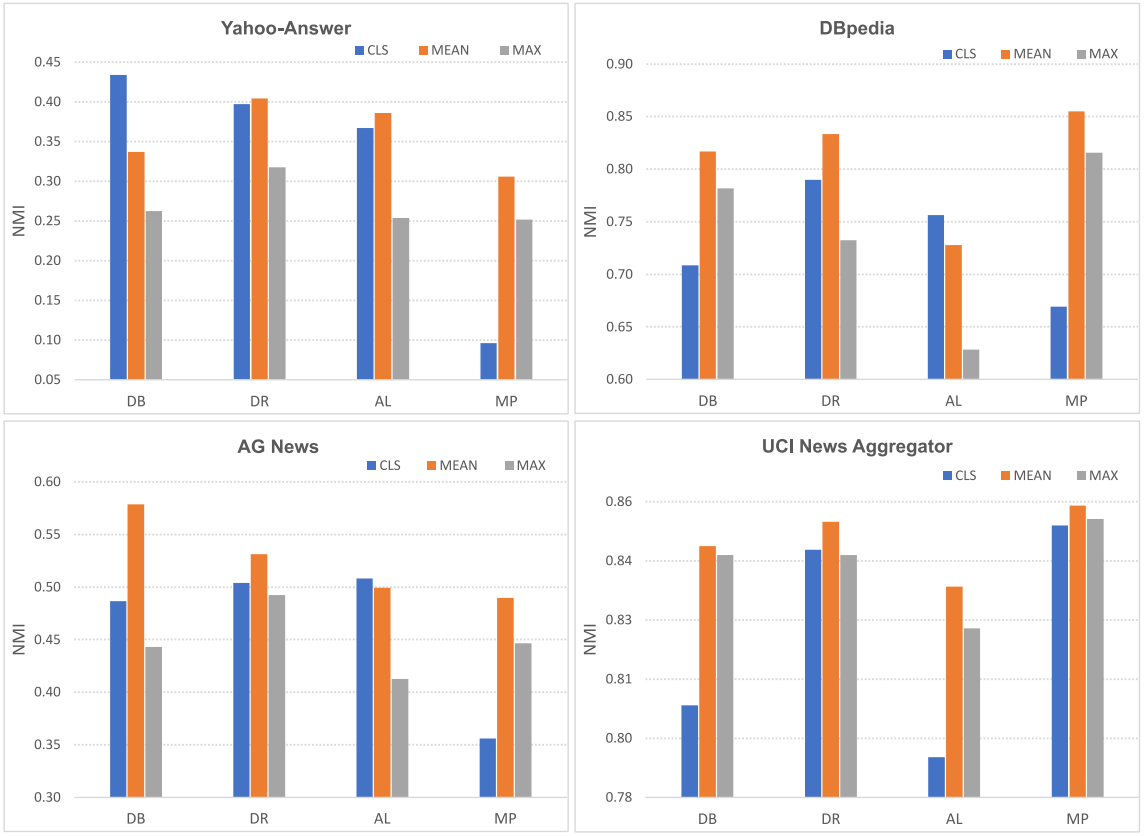

Fig. 4. Comparison of clustering results across SBERT models on each dataset utilizing the NMI metric for different pooling techniques; DB:DistilBERT, DR:DistilRoBERTa, AL:ALBERT, MP:MPNET.

are outlined in Table 7, detailing embedding generation followed by clustering with K-Means. When comparing the SBERT models, DistilBERT, DistilRoBERTa, and ALBERT generate embeddings significantly faster than MPNET due to their distilled nature, offering improved computational efficiency.

On the other hand, using SBERT models in their raw form, without specific training for this study, highlights their robust transfer learning capability in text clustering. Fine-tuning these models with similar datasets will likely improve their performance. Furthermore, our method has a significant advantage over existing approaches as it does not require additional hyperparameter tuning, making it simpler to implement. Another noteworthy advantage of our approach is that it is not limited to clustering only short text, unliked the compared studies. It can also be effectively used in long text clustering since it produces a single sentence embedding for the entire text.

4.5. Discussion and implications

The consistent and robust clustering performance observed across all four datasets and the model's ability to compete with existing methods in the literature indicate that SBERT models could be employed as a transfer learning tool in text clustering endeavors. In particular, the clustering performances on DBpedia and UCI News Aggregator almost match their grand-truth values. In order to evaluate the performance of individual models, we calculated the average of 12 NMI results derived from each model (as presented in Tables 3, 4, 5, and 6). This approach allows for an unbiased evaluation of the models, regardless of the pooling techniques and datasets used. The calculated averages give the following hierarchy of model success: DistilRoBERTa (0.628), DistilBERT (0.611), ALBERT (0.582), and MPNET (0.570).

In the context of assessing the compatibility of models with pooling techniques, regardless of datasets, we performed another analysis

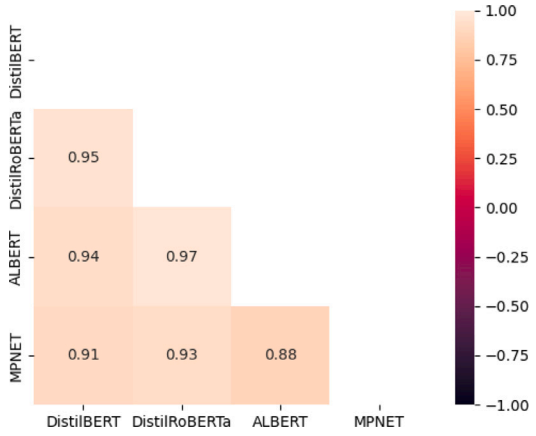

Fig. 5. Correlation matrix of SBERT models, including all metrics, all pooling techniques, and all datasets.

by computing the average of the three NMI results gathered from each pooling technique and SBERT model pairs across Tables 3, 4, 5, and 6. For the CLS pooling, the ranking of success is established as follows: DistilRoBERTa (0.633), DistilBERT (0.608), ALBERT (0.605), and MPNET (0.493). Under the Mean pooling, the order of success unfolds as: DistilRoBERTa (0.655), DistilBERT (0.644), MPNET (0.626), and ALBERT (0.612). For the max pooling, the hierarchy of success materializes as: DistilRoBERTa (0.596), MPNET (0.591), DistilBERT (0.582), and ALBERT (0.529).

Both experiments above suggest that DistilRoBERTa is more successful than the other models, particularly in terms of the NMI measure. In these experiments, ARI, HOM, and COMP metrics closely mirror

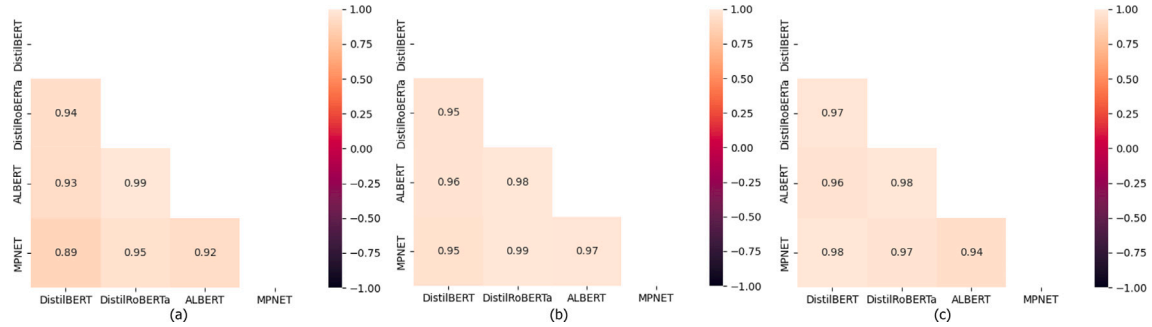

Fig. 6. Correlation matrices of SBERT models for each pooling technique including all metrics and all datasets, (a) CLS, (b) Mean, (c) Max.

Table 8
The statistical results of ANOVA-test.

Abstract						
Groups	*Count*	*Sum*	*Average*	*Variance*		
CLS	64	34.542	**0.540**	0.044		
Mean	64	37.826	**0.591**	0.042		
Max	64	33.378	**0.527**	0.050		
ANOVA						
Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.147	2	0.073	1.612	**0.202**	3.044
Within Groups	8.591	189	0.045			
Total	8.737	191				

those of the NMI metric. Therefore, no separate individual analysis was conducted for ARI, HOM, and COMP in the previous evaluations. The compatibility in outputs across these metrics underscores the consistency and reliability of the clustering performance evaluations.

Fig. 5 presents the correlation matrices of all models with respect to NMI, ARI, HOM ve COMP metrics. The matrix underscores the absence of significant differences in clustering performance among the models when considering the results from all datasets. Only ALBERT and MPNET have a slightly lower correlation compared to the others.

Fig. 6 provides deeper insights into model correlations, grouping the results based on pooling techniques. Specifically, Figs. 6.a, 6.b, and 6.c offer correlation matrices of the model for the CLS, Mean, and Max pooling techniques, respectively. Three correlation matrices support the conclusion that all models show similar trends in each pooling method. The findings and former results suggest that although DistilRoBERTa performs slightly better than other models, there is no significant superiority among them. This implication is also supported by the observation that DistilRoBERTa is not the top-performing model for any of the datasets, as illustrated in Fig. 4.

Examining the harmony between pooling techniques and models in the light of the graphs in Fig. 4, DistilRoBERTa and MPNET consistently achieve peak performance with the Mean pooling technique across all datasets. DistilBERT attains the best results with the Mean technique in three out of the four datasets. Contrarily, ALBERT exhibits a slightly different behavior, delivering optimal outcomes with the Mean pooling in the Yahoo Answers and UCI News Aggregator datasets, while getting its highest results with the CLS technique in the other two datasets. Based on these findings, it can be concluded that, in general, the models tend to demonstrate superior performance with mean pooling. On the other hand, it is not feasible to determine the superiority of any particular model+pooling technique pair over others. For instance, the DistilBERT+CLS pair yielded the most successful result in the Yahoo-Answer dataset, MPNET+Mean in DBPedia, DistilBERT+Mean in AG News, and MPNET+Mean in UCI News Aggregator. These results imply that the clustering performance of model+pooling method pairs varies depending on the dataset.

We also conducted a supplementary test to determine whether there is a statistically significant difference between the pooling techniques applied to models. In this context, we applied a one-way ANOVA test by grouping each pooling technique's results, including all models and all metric values. The null hypothesis of ANOVA is that there is no significant difference between the pooling methods. The alternative hypothesis is that at least one pooling method is significantly different from the overall average regarding clustering metrics. We defined the significance level (α) of the hypothesis test as 0.05. Table 8 displays the ANOVA test results. With a p-value of 0.202 (> 0.05) exceeding the α reference value, it is deduced that there is no significant difference between the pooling techniques. However, when examining the average values of the pooling methods in the abstract of Table 8, it is evident that the order of success is as Mean, CLS, and Max. This finding and the previous results indicate that the Mean method is more preferable to the others, although it does not provide a great superiority.

So far this paper has demonstrated the theoretical and experimental effectiveness of SBERT-based models in text clustering. Additionally, these models have practical applications in a variety of real-world scenarios. Here are some important cases where these models provide potential value:

- In large document repositories, SBERT-based clustering enables the grouping of semantically similar documents, even without exact keyword matches. This streamlines information retrieval and improves document discoverability.
- SBERT models can cluster social media content to reveal trending topics and overall sentiment, offering invaluable insight for social engineering efforts.
- Media organizations can use our approach to cluster news articles based on semantic similarity. Thus, they can provide readers personalized news feeds tailored to their interests and preferences.
- By clustering product descriptions, customer reviews, and inquiries, SBERT models help e-commerce companies extract valuable information. These insights lead to improvements in various aspects, including product development, marketing strategies, and customer support, ultimately increasing customer satisfaction and loyalty.

5. Conclusion

Recently, LLMs have received substantial attention in text clustering as well as in many other areas of NLP. In particular, the numerical text representations produced by these models can be used as features by clustering algorithms. Compared to other text digitization methods, LLMs create more semantically and contextually realistic digital representations of texts. However, these numerical representations are for general-usage purposes and frequently not suitable for text similarity comparison and clustering. To overcome this problem, SBERT models have been proposed that generate sentence embeddings that represent the entire text in a holistic approach rather than a word-based approach. As the research on text clustering applications with these models remains relatively sparse in the existing literature, this study focuses on a comprehensive analysis of SBERT models's transfer learning capabilities in text clustering. In addition, the performance of

these models in text clustering is compared with state-of-the-art results in the literature. Furthermore, this study presents a further analysis by applying distinct pooling techniques to these models, to measure the impact of these techniques on the models.

In this context, CLS, Mean and Max pooling methodologies were applied to DistilBERT, DistilRoBERTa, ALBERT and MPNET models across the four distinct corpora. The results highlight that none of these models outperforms the other in all corpora. Nevertheless, in terms of pooling techniques, we conclude a general consensus: Mean pooling outperforms the others. Subsequently, the success hierarchy is established, with CLS following as the next favorable technique, while Max demonstrates the lowest performance levels. On the other hand, although such a ranking is obtained, our findings indicate that there are no significant differences in the results produced by these techniques. Consequently, with the exception of ALBERT, all other SBERT models tend to perform more effectively when combined with the Mean pooling technique.

One limitation of this study is the requirement to specify the number of clusters in advance. To address this issue, as a future plan, a hybrid clustering method that combines K-means with meta-heuristic optimization algorithms will be developed. This approach can provide efficient clustering solutions that automatically determine the optimal number of clusters. In the preliminary studies using Particle Swarm Optimization (PSO), it was observed that PSO was not ideal for 768-dimensional sentence embeddings. However, promising results were achieved when dimensionality reduction techniques, such as Principal Component Analysis (PCA) or t-Distributed Stochastic Neighbor Embedding, were employed to reduce embeddings to the 10–50 dimension range. The future work involves defining the optimum dimension reduction technique and evaluating the performance of several meta-heuristic algorithms combined with K-means to automatically determine the optimal number of clusters.

Funding statement

The author states that there is no funding organization for this study.

Ethics approval statement

The author states that ethical approval is not applicable for this study.

CRediT authorship contribution statement

Yasin Ortakci: Conceptualization, Data curation, Investigation, Methodology, Software, Writing – original draft.

Declaration of competing interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All datasets used in this study are publicly available and their links are shared in the manuscript.

References

[1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.
[2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019, arXiv preprint arXiv:1907.11692.
[3] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving Language Understanding by Generative Pre-Training, OpenAI, 2018.
[4] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang, Pre-trained models for natural language processing: A survey, Sci. China Technol. Sci. 63 (10) (2020) 1872–1897.
[5] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R.R. Salakhutdinov, Q.V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, Adv. Neural Inf. Process. Syst. 32 (2019).
[6] A.G. d'Sa, I. Illina, D. Fohr, Bert and fasttext embeddings for automatic detection of toxic speech, in: 2020 International Multi-Conference on:"Organization of Knowledge and Advanced Technologies", OCTA, IEEE, 2020, pp. 1–5.
[7] N. Reimers, B. Schiller, T. Beck, J. Daxenberger, C. Stab, I. Gurevych, Classification and clustering of arguments with contextualized word embeddings, 2019, arXiv preprint arXiv:1906.09821.
[8] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI Blog 1 (8) (2019) 9.
[9] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 33 (2020) 1877–1901.
[10] W. Alhoshan, A. Ferrari, L. Zhao, Zero-shot learning for requirements classification: An exploratory study, Inf. Softw. Technol. 159 (2023) 107202.
[11] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, 2019, arXiv preprint arXiv:1908.10084.
[12] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2014, pp. 1532–1543.
[13] J. Ghosh, A. Strehl, Similarity-based text clustering: A comparative study, in: Grouping Multidimensional Data: Recent Advances in Clustering, Springer, 2006, pp. 73–97.
[14] D. Yan, K. Li, S. Gu, L. Yang, Network-based bag-of-words model for text classification, IEEE Access 8 (2020) 82641–82652.
[15] L.H. Patil, M. Atique, A novel approach for feature selection method TF-IDF in document clustering, in: 2013 3rd IEEE International Advance Computing Conference, IACC, IEEE, 2013, pp. 858–862.
[16] P. Bafna, D. Pramod, A. Vaidya, Document clustering: TF-IDF approach, in: 2016 International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT, IEEE, 2016, pp. 61–66.
[17] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, 2013, arXiv preprint arXiv:1301.3781.
[18] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Trans. Assoc. Comput. Linguist. 5 (2017) 135–146.
[19] A. Subakti, H. Murfi, N. Hariadi, The performance of BERT as data representation of text clustering, J. Big Data 9 (1) (2022) 1–21.
[20] Y. Li, J. Cai, J. Wang, A text document clustering method based on weighted bert model, in: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference, Vol. 1, ITNEC, IEEE, 2020, pp. 1426–1430.
[21] F. Jáñez-Martino, R. Alaiz-Rodríguez, V. González-Castro, E. Fidalgo, E. Alegre, Classifying spam emails using agglomerative hierarchical clustering and a topic-based approach, Appl. Soft Comput. 139 (2023) 110226.
[22] A. Moura, P. Lima, F. Mendonça, S.S. Mostafa, F. Morgado-Dias, On the use of transformer-based models for intent detection using clustering algorithms, Appl. Sci. 13 (8) (2023) 5178.
[23] B. Yin, M. Zhao, L. Guo, L. Qiao, Sentence-BERT and k-means based clustering technology for scientific and technical literature, in: 2023 15th International Conference on Computer Research and Development, ICCRD, IEEE, 2023, pp. 15–20.
[24] H. Choi, J. Kim, S. Joe, Y. Gwon, Evaluation of bert and albert sentence embedding performance on downstream nlp tasks, in: 2020 25th International Conference on Pattern Recognition, ICPR, IEEE, 2021, pp. 5482–5487.
[25] X. Yan, J. Guo, S. Liu, X. Cheng, Y. Wang, Learning topics in short texts by non-negative matrix factorization on term correlation matrix, in: Proceedings of the 2013 SIAM International Conference on Data Mining, SIAM, 2013, pp. 749–757.
[26] X. Yan, J. Guo, S. Liu, X.-q. Cheng, Y. Wang, Clustering short text using ncut-weighted non-negative matrix factorization, in: Proceedings of the 21st ACM International Conference on Information and Knowledge Management, 2012, pp. 2259–2262.
[27] X. Yan, J. Guo, Y. Lan, X. Cheng, A biterm topic model for short texts, in: Proceedings of the 22nd International Conference on World Wide Web, 2013, pp. 1445–1456.
[28] J. Yin, J. Wang, A dirichlet multinomial mixture model-based approach for short text clustering, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 233–242.

[29] J. Chen, Z. Gong, W. Liu, A nonparametric model for online topic discovery with word embeddings, Inform. Sci. 504 (2019) 32–47.
[30] J. Kumar, J. Shao, R. Kumar, S.U. Din, C.B. Mawuli, Q. Yang, A context-enhanced Dirichlet model for online clustering in short text streams, Expert Syst. Appl. 228 (2023) 120262.
[31] L. Akritidis, M. Alamaniotis, A. Fevgas, P. Bozanis, Confronting sparseness and high dimensionality in short text clustering via feature vector projections, in: 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence, ICTAI, IEEE, 2020, pp. 813–820.
[32] L. Akritidis, M. Alamaniotis, A. Fevgas, P. Tsompanopoulou, P. Bozanis, Improving hierarchical short text clustering through dominant feature learning, Int. J. Artif. Intell. Tools 31 (05) (2022) 2250034.
[33] M.H. Ahmed, S. Tiun, N. Omar, N.S. Sani, Short text clustering algorithms, application and challenges: A survey, Appl. Sci. 13 (1) (2022) 342.
[34] B.A.H. Murshed, S. Mallappa, J. Abawajy, M.A.N. Saif, H.D.E. Al-Ariki, H.M. Abdulwahab, Short text topic modelling approaches in the context of big data: taxonomy, survey, and analysis, Artif. Intell. Rev. 56 (6) (2023) 5133–5260.
[35] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.
[36] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).
[37] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, S. Fidler, Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 19–27.
[38] Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., Google's neural machine translation system: Bridging the gap between human and machine translation, 2016, arXiv preprint arXiv:1609.08144.
[39] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, 2019, arXiv preprint arXiv:1910.01108.
[40] R. Sennrich, B. Haddow, A. Birch, Neural machine translation of rare words with subword units, 2015, arXiv preprint arXiv:1508.07909.
[41] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: A lite bert for self-supervised learning of language representations, 2019, arXiv preprint arXiv:1909.11942.
[42] K. Song, X. Tan, T. Qin, J. Lu, T.-Y. Liu, Mpnet: Masked and permuted pre-training for language understanding, Adv. Neural Inf. Process. Syst. 33 (2020) 16857–16867.
[43] J. Haas, W. Yolland, B. Rabus, Simple high quality OoD detection with L2 normalization, 2023, arXiv preprint arXiv:2306.04072.
[44] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.
[45] L. Pugachev, M. Burtsev, Short text clustering with transformers, 2021, arXiv preprint arXiv:2102.00541.
[46] X. Zhang, J. Zhao, Y. LeCun, Character-level convolutional networks for text classification, Adv. Neural Inf. Process. Syst. 28 (2015).
[47] A. Amelio, C. Pizzuti, Correction for closeness: Adjusting normalized mutual information measure for clustering comparison, Comput. Intell. 33 (3) (2017) 579–601.
[48] J.E. Chacón, A.I. Rastrojo, Minimum adjusted rand index for two clusterings of a given size, Adv. Data Anal. Classif. 17 (1) (2023) 125–133.
[49] I. Pauletic, L.N. Prskalo, M.B. Bakaric, An overview of clustering models with an application to document clustering, in: 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO, IEEE, 2019, pp. 1659–1664.