Resume Shortlisting and Ranking with Transformers

Vinaya James(✉), Akshay Kulkarni, and Rashmi Agarwal

REVA Academy for Corporate Excellence (RACE), REVA University, Bangalore, India
{vinayajames.AI01,akshaykulkarni}@race.reva.edu.in,
rashmi.agarwal@reva.edu.in

Abstract. The study shown in this paper helps the human resource domain eliminate the time-consuming recruitment process task. Screening resume is the most critical and challenging task for human resource personnel. Natural Language Processing (NLP) techniques are the computer's ability to understand spoken/written language. Now a day's, online recruitment platform is more vigorous along with consultancies. A single job opening will get hundreds of applications. To discover the finest candidate for the position, Human Resource (HR) employees devote extra time to the candidate selection process. Most of the time, shortlisting the best fit for the job is time-consuming and finding an apt person is hectic. The proposed study helps to shortlist the candidates with a better match for the job based on the skills provided in the resume. As it is an automated process, the candidate's personalized favor and soft skills are not affected by the hiring process. The Sentence-BERT (SBERT) network is a Siamese and triplet network-based variant of the Bidirectional Encoder Representations from Transformers (BERT) architecture, which may generate semantically significant sentence embeddings. An end-to-end tool for the HR domain, which takes hundreds of resumes along with required skills for the job as input and provides the better-ranked candidate fit for the job as output. The SBERT is compared with BERT and proved that it is superior to BERT.

Keywords: Natural Language Processing · Sentence-BERT · Automatic Recruitment Process · Sentence Embedding

1 Introduction

In a business or organization, it is indeed critical to make the proper hiring decisions for particular positions for Human Resources Manager or Head-hunter [1]. Recruitment tools like "LinkedIn" and "Monster" search for skills and identify candidates who are qualified for open positions. The list of resume search results may be lengthy. All resumes should be manually reviewed to identify possible applicants. Especially, large companies like "Google" frequently receive hundreds of thousands of resumes each year for job applications. As a result, automation is introduced to make the work easy with time-saving.

S. Nandan Mohanty et al. (Eds.): ICISML 2022, LNICST 471, pp. 99–108, 2023.
https://doi.org/10.1007/978-3-031-35081-8_8

The introduction of NLP simplifies the process of automatic sentence or test recognition from papers or resumes [2]. In general vector, representations have become an ever-existing entity in the NLP context. Word embedding properties have seen major developments in the recent past, such as the superposition of word contexts [3], pointwise linkages to shared data values between related words, and linear substructures [4]. However, locating sentences that are comparable based on context, meaning, subject, etc. is a problem in naturally occurring language processing, which leads to the issue of statement integration. The text can be organized, produced, processed, etc., and based on this, it converts the vectors of the words into a vector representation of the sentence. In contrast, an embedding definition is a numerical representation of a word, phrase, sentence, or longer natural language utterance in a particular space. Additionally, word embeddings distributed semantic vector representations of words have drawn a lot of interest in recent years and have undergone a lot of changes [5]. The word embeddings were developed by analyzing the word distribution among arbitrary text data, and because they are familiar with word semantics, they are a crucial part of many semantic similarity algorithms.

Compared to word embeddings, sentences encompass a higher level of information: the complete semantics and syntactic declarations of the sentence. The word2vec model, which is one of the most popular models, employs neural networks to create word embeddings [6]. Recent transformer-based models' pre-trained word embeddings provided cutting-edge outcomes in a variety of NLP tasks, including semantic similarity. Based on this, modern language models like RoBERTa, BERT [7, 8], and ALBERT use transformers to traverse the underlying corpus in both directions and create vector representations of text data. In 2009, the BERT transformer model outperformed the best results available for different NLP tasks, including semantic similarity. The BERT model is typically tuned to serve a specific NLP task using labeled training data. Multilingual BERT is enhanced by the alignment method suggested by Cao et al. [9]. Thus, the BERT models produce bilingual illustrations of words that consider the word's context in both directions.

Performing sentence pair regression, like clustering and semantic search [10], can be time-consuming with BERT due to the large corpus size. Converting a sentence into a vector that encodes the semantic meaning of the sentence is one efficient technique to address this issue. Transformers are the guide to a state-of-art model, and the next section discusses more works accomplished on the NLP concept.

2 State of Art

As sentence embeddings are more realistic representations of text, recent papers on sentence embedding transformers have focused on the mechanics of recognition. Fast Text is a model that incorporates a character-based n-gram model in addition to the established word embedding techniques Word2Vec and GloVe [11]. This makes it possible to calculate word embeddings from the vocabulary.

Vaswani et al. [12] suggested the Transformer, a self-attention network, as a remedy for the neural sequence-to-sequence issue. When a self-attention network is used to visualize a phrase, each word is represented by a scaled sum of all the other words in the phrase. Liu et al. [13] used a sentence's inner attention to show that self-attention pooling existed before self-attention networks. By generalizing scalar attention to vectors, Cho et al. [14] devised a fine-grained attention approach for neural machine translation. Natural Language Inference (NLI) and other supervised transfer tasks can help complex phrase encoders that are often pre-trained like language models. The Universal Sentence Encoder improves unsupervised learning by training a transformer network on the Stanford NLI (SNLI) dataset. According to Giorgiet al. [15], the task that sentence embeddings are trained on greatly influences their quality. The SNLI datasets, according to Conneau et al. [16], are appropriate for training sentence embeddings.

Therefore, to successfully develop the Chinese Depth Approximation Networking (DAN) and transformers, Parameswa et al. [17] offer an approach that uses Internet responses. Naseem et al. [18] fed the static word embeddings GloVe through the deep neural networks and carried out a controlled neural interpretation operation to obtain the perspective encoding. The Remiers et al. [19] ELMo model is a method for a deeper summarized description. The internal states of words are used to teach a deep bidirectional Language Model (biLM) that has already been trained on a large text corpus. Pre-trained language models of the fine-tuning variety have unfrozen pretrained parameters that can be changed for a new assignment. The text presentation is no longer extracted using this approach. Two significant tasks were introduced by Devlin et al. [20] in their autoencoding pre-trained language model BERT, a deep bidirectional Transformers model: Mask Language Model (MLM) and Next Sentence Prediction (NSP). During the tuning stage, the only activities that deviate from one another are those at the input and output layers. Dai, Zihang, et al. [21] proposed the Transformer-XL generalized autoregressive pre-trained language model, which can learn bidirectional contexts by maximizing expected likelihood across all possible factorization order permutations. Additionally, two groups of the pre-trained language model can be identified based on the pre-training procedure: Auto Encoding (AE) and Auto-Regressive (AR). AR language models like ELMo and XLNet aim to estimate the probability distribution of a text corpus by employing an autoregressive model. However, AE language models like BERT and its variations like RoBERTa by Liu, Yinhan, et al. [22] seek to recreate the original data from corrupted input without resorting to explicit density estimates.

One major problem with BERT-type models is the introduction of fictitious symbols like MASK during pretraining, despite their complete absence from the final output text. The pre-trained model effectively illustrates how words or phrases link to one another and access various data. Thus, embedding-based key phrase extraction has recently demonstrated strong performance. Lee et al. [23] suggested a Deep Belief Network (DBN) to model the hierarchical relationship between key phrase embeddings. It is simple to distinguish the target document from others using this strategy. Reference Vector Algorithm (RVA) by Papagiannopoulou et al. [24] is a key phrase extraction technique that uses local word vectors as a guiding principle, employing an average of the embeddings trained on distinct files using GloVe as the reference vector for all candidate phrases. The score used to rank the candidate key phrases is then determined by calculating how closely the embeddings of each candidate key phrase match those of the reference vector.

Bennani-Somerset et al. proposed EmbedRank [25] based on the cosine similarity between the candidate key phrase's embeddings and the document's sentence embeddings. EmbedRank creates a document representation using the phrase embedding models Doc2Vec and Sent2Vec. They employ Maximal Marginal Relevance (MMR) to further broaden the keyword's applicability. Pre-trained language models, in particular, BERT, have recently gained popularity and significantly enhanced performance for several NLP applications. Pre-trained BERT and its variations have mostly proved successful in the English language. A language-specific model might be retrained using the BERT architecture for other languages, or one could employ pre-trained multilingual BERT-based models.

3 Problem Definition

Identifying a qualified candidate for the job is a complex undertaking. Typically, manual processes are used in the traditional hiring process. The HR department's qualified recruiters and other significant resources are needed for the manual recruitment process. Businesses sometimes receive a substantial volume of resumes for each job opening, some of which may not even be suitable for the position. Additionally, these hiring procedures take a lot of time and effort to discover qualified applicants for open positions.

Therefore, manually selecting the most pertinent applicants from a lengthy list of potential candidates is difficult. Numerous recent research has focused on the drawbacks of the manual hiring process. Dealing with resumes in the advertising of job specifications and hiring procedures. Selecting people who fit a given job profile is a vital task for most firms. As online hiring becomes more common, traditional hiring methods become less successful.

Recruiting the most pertinent multilingual candidates through the manual hiring process is one of the most critical issues in multilingual job offers and resumes.

The proposed model investigates developing a resume shortlisting and ranking with Transformers to address the difficulty of selecting the right candidate out of two hundred resumes. An automated recruiting system is essential to make it easier for job seekers to access recruitment opportunities and to minimize the amount of human effort involved in the hiring process.

The main goal is to improve the current resume ranking scheme and make it more adaptable for both parties.

1. *Those who were hired as candidates:* Candidates who have recently graduated and are looking for a job. A significant portion of those applicants is so desperate that they are willing to work in any position unrelated to their skill set and abilities.
2. *The client organization that recruits the applicants:* A job recruiter's objective is to select the top candidate from all qualified applicants based on the Job Description (JD). The process takes time for both the individual and the business. With an automatic resume sorting system, the business may produce the finest candidate list possible based on the limits and requirements they gave for that particular role. Since the appropriate person will be hired for the position, this hiring method will benefit both

the candidate and the organization. Therefore, neither the client firm nor the hired candidate would have any regrets.

The three objectives of this study are:

1. *Collect the resumes as per the defined JD:* The resumes are collected by HR from online job platforms, referrals from existing employees, and third-party consultancies. But getting the exact JD-related resumes are challenging. For this model two hundred resumes have been collected.
2. *Build a custom algorithm to shortlist the resume as per the JD given:* The skills are extracted to a *pandas* data frame with the help of *ResumeParser*. The resume matches will be shortlisted and added to a list based on the JD that HR has provided.
3. Create a ranking *algorithm to get the best out of shortlisted resumes:* JDs and candidates' skill sets are compared, and the most suitable individual with the necessary skills is shortlisted. The model is trained using SBERT and BERT model encodings, and the top N resumes are sorted according to their cosine similarity to sentences and words, respectively.

4 Proposed Method

Finding the "right" applicant for a position has never been simple. In addition to having the required training and work experience, a prospective employee typically needs to fit in with the current team and support the company's vision. A new dilemma has emerged with the rise of internet job boards and globalization. Today's recruiting professionals frequently need to analyze hundreds of online profiles and resumes only to pick whom to approach due to how simple it has become to build an online profile and apply to a position with a few clicks.

It is not surprising that various technology solutions have been offered to aid recruiters in addressing the problem of candidate screening because automating the shortlisting of candidates can lead to lower costs and higher recruiter productivity. To rank the resume successfully, an efficient context test-based embedding is needed. To achieve textual similarity, transfer-based models like BERT and XLNet are used. In addition, an efficient pre-trained model is required due to its poor accuracy. SBERT, a version of the BERT network that can produce semantically significant sentence embeddings by combining Siamese and triplet networks, is employed in this way [26]. The suggested architecture is depicted in Fig. 1.

The number of resumes is the input used in this proposed model methodology to shortlist the most suitable individuals. *ResumeParser* is used to extract the required details of jobseekers, like Name, Mobile Number, Email Address, and Skills. Exploratory Data Analysis (EDA) is used to remove duplicates, locate databases, and predict and remove missing or null values from text or sentences in resumes. The next step is data preprocessing. At this step, improper words are eliminated, the text is normalized, and the words are prepared for further processing with stop words, Stemming, Lemmatization, and Latent Dirichlet Allocation (LDA). Making vector representations of all words and documents and collectively embedding them in a common vector space is the first step in the extraction of relevant skills and expertise. Then SBERT and BERT are used to create a model.

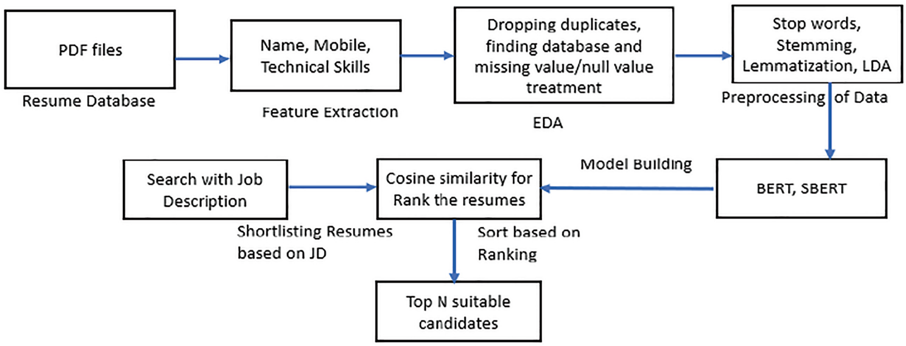

Fig. 1. Proposed Model

The produced embedding vectors can be subjected to the Semantic Textual Similarity (STS) task employing metrics such as the Manhattan distance, Cosine similarity, and Euclidean distance. BERT, Siamese networks, and Pooling layer are the three key principles that the SBERT architecture employs. Using a pre-training method called MLM, which typically masks a portion of the tokens in a sentence and predicts them based on their context. BERT is a deep bidirectional transformer because MLM predicts that tokens will approach it from both directions. This is different from traditional left-to-right models, which only work one way. The JD is then matched with terms described in skills to get a cosine similarity score that may be used to rank and shortlist candidates appropriately in the HR field.

5 Modeling

The term “dictionary” refers to a list of distinct words for each category of documents after the stop words have been eliminated and the words have been stemmed. These stemmed nonstop words can easily address several fields in a database by adding field names to them. Both words with several grammatical forms and words with similar meanings derived from them are handled by an algorithm. Taking all these stages into account, the techniques employed in the example are:

Step 1: Divide the text into words.

Step 2: Eliminate all punctuation and symbols and, if desired, lowercase all words.

Step 3: Eliminate the stop words.

Step 4: Use the Snowball Stemming Algorithm to stem the words.

Step 5: Add parenthesis to each word before adding the field names (if appropriate).

Two hundred resumes are gathered from various sources as part of the data collection process. All the necessary features, including Name, Email address, Mobile Number, and abilities, are retrieved with the aid of *ResumeParser.* A model for BERT and SBERT is constructed based on the JD following EDA and data preprocessing. The *N* number of resumes with the highest ranking is listed using the cosine similarity between the skill set and required JD. The proposed software design is described in Fig. 2.

SBERT for the STS task permits two steps in the prediction of similarity:

Step 1: First, using a sentence encoder, obtain sentence embeddings for each sentence.

Step 2: Next, as the model-predicted similarity SBERT and BERT, compute the cosine similarity between the two embeddings of the input sentence. SBERT used *the bert-base-nli-mean-tokens* model and is compared to the BERT *bert-base-uncased* model.

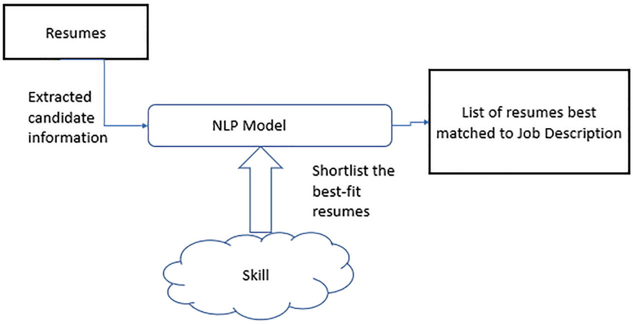

Fig. 2. Software Design

6 Analysis and Results

Sentence embedding that outperforms the Classical Least Squares (CLS) vector is obtained by the average of the SBERT context embedding's one or two layers. The average of the last two layers of SBERT, denoted by *last2avg*, consistently yields better results than the average of the last layer of BERT. Since unsupervised learning methods like topic modeling do not guarantee that their results can be understood, correlation metrics have gained popularity among text-mining experts.

The degree of semantic similarity among top-ranking terms in each topic is measured by correlativity. It determines the co-occurrence scores of words in the modeled documents. As coherence also works with syntactic information with the aid of a sliding window that traverses across the corpus and checks occurrences. The notion behind coherence calculation is strongly related to embedding representations of text. Different methods can be used to calculate the correlation. SBERT gives a better solution than BERT when a comparison of top ten ranked resumes based on JD.

Table 1 shows the consistency and alignment of different sentence embedding models (BERT and SBERT) and their averaged STS results. Among BERT and SBERT, models with superior alignment and homogeneity outperform in comparison with the models which do not have alignment and homogeneity. Scores for each pair of sentences are calculated by applying the cosine similarity scoring function to the sentence vectors. The SBERT method is used to arrange the sentences to maximize the sum of their similarities. The findings of this method show that the SBERT method always performs better than the BERT method.

Table 1. Coherence value comparison for BERT and SBERT

Data Set	Model	Correlation value for Similarity
STS1	SBERT	0.42649
	BERT	0.194206
STS2	SBERT	0.378602
	BERT	0.119996
STS3	SBERT	0.377433
	BERT	0.047986
STS4	SBERT	0.374302
	BERT	0.156387
STS5	SBERT	0.373682
	BERT	0.182748
STS6	SBERT	0.373111
	BERT	0.048559

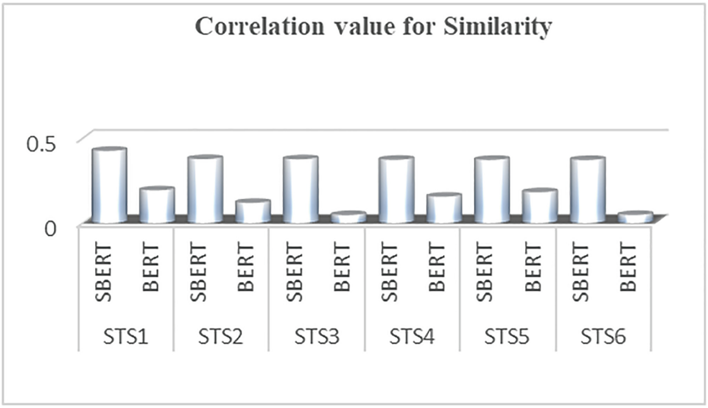

Fig. 3. Performance analysis of SBERT with Coherence

Based on Table 1, the graphical representation is provided in Fig. 3.

Table 1 analysis reveals that the SBERT performs better than the BERT in terms of correlation. Compared to SBERT, the similarity variation is lower, as depicted in Fig. 3. Resumes in the STS3 dataset exhibit a relatively low similarity variance. As a result, the SBERT executes the ideal sentence embedding to rank the candidate's information with the JD of the job provider in the minimum similarity variance.

7 Conclusion and Future Works

The proposed SBERT transformer helps recruiters screen resumes more quickly and effectively, cutting the cost of hiring. Thus, the company will then have access to a potential applicant who will be successfully placed in a business that appreciates the candidate's skills and competencies. These days, many applicants submit applications for interviews. Every interview should include a review of resumes. Going through each resume one by one is not a good idea. It becomes quite difficult for the HR team to narrow down candidates for the following stage of the hiring process. The SBERT streamlines the process by summarizing resumes and classifying them by how closely they match the organization's necessary skills and requirements.

The proposed method evaluates candidates' skills and ranks them by the JD and skill requirements of the employing organization. A summary of their resume is supplied to provide a fast overview of each candidate's qualifications. One of the main issues is when a candidate lists skills for which they have no experience because the model focuses on the skill set listed on the resume submitted by the candidate. Artificial Intelligence techniques or any other effective sentence embedding transformers can be used for further improvement.

References

1. Siddique, C.M.: Job analysis: a strategic human resource management practice. Int. J. Human Resour. Manage. **15**(1), 219–244 (2004)
2. Sanabria, R., et al.: How2: a large-scale dataset for multimodal language understanding. arXiv preprint arXiv:1811.00347 (2018)
3. Arora, S., Li, Y., Liang, Y., Ma, T., Risteski, A.: A latent variable model approach to PMIbased word embeddings. Trans. Assoc. Comput. Linguist. **4**, 385–399 (2016)
4. Rieck, B., Leitte, H.: Persistent homology for the evaluation of dimensionality reduction schemes. Comput. Graph. Forum **34**(3) (2015)
5. Stein, R.A., Jaques, P.A., Valiati, J.F.: An analysis of hierarchical text classification using word embeddings. Inf. Sci. **471**, 216–232 (2019)
6. Mishra, M.K., Viradiya, J.: Survey of sentence embedding methods. Int. J. Appl. Sci. Comput. **6**(3), 592 (2019)
7. Chernyavskiy, A., Ilvovsky, D., Nakov, P.: Transformers: "The End of History" for natural language processing? In: Oliver, N., Pérez-Cruz, F., Kramer, S., Read, J., Lozano, J.A. (eds.) ECML PKDD 2021. LNCS (LNAI), vol. 12977, pp. 677–693. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86523-8_41
8. Suryadjaja, P.S., Mandala, R.: Improving the performance of the extractive text summarization by a novel topic modeling and sentence embedding technique using SBERT. In: 2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA), pp. 1–6 (2021). https://doi.org/10.1109/ICAICTA53211.2021.9640295
9. Cao, S., Kitaev, N., Klein, D.: Multilingual alignment of contextual word representations. arXiv preprint arXiv:2002.03518 (2020)
10. Choi, H., Kim, J., Joe, S., Gwon, Y.: Evaluation of BERT and ALBERT sentence embedding performance on downstream NLP tasks. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 5482–5487 (2021). https://doi.org/10.1109/ICPR48806.2021.9412102

11. Dharma, E.M., et al.: The accuracy comparison among word2vec, glove, and fasttext towards convolution neural network (CNN) text classification. J. Theor. Appl. Inf. Technol. **100**(2), 31 (2022)
12. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
13. Liu, Y., Sun, C., Lin, L., Wang, X.: Learning natural language inference using bidirectional LSTM model and inner-attention. arXiv preprint arXiv:1605.09090 (2016)
14. Choi, H., Cho, K., Bengio, Y.: Fine-grained attention mechanism for neural machine translation. Neurocomputing **284**, 171–176 (2018). ISSN 0925-2312
15. Giorgi, J., et al.: DeCLUTR: deep contrastive learning for unsupervised textual representations. arXiv preprint arXiv:2006.03659 (2020)
16. Conneau, A., et al.: Supervised learning of universal sentence representations from natural language inference data. arXiv preprint arXiv:1705.02364 (2017)
17. Parameswaran, P., Trotman, A., Liesaputra, V., Eyers, D.: Detecting the target of sarcasm is hard: really?? Inf. Process. Manage. **58**(4), 102599 (2021)
18. Naseem, U., Musial, K.: DICE: deep intelligent contextual embedding for Twitter sentiment analysis. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 953–958 (2019). https://doi.org/10.1109/ICDAR.2019.00157
19. Reimers, N., Gurevych, I.: Alternative weighting schemes for elmoembeddings. arXiv preprint arXiv:1904.02954 (2019)
20. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long and Short Papers), Minneapolis, Minnesota, pp. 4171–4186. Association for Computational Linguistics (2019)
21. Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q.V., Salakhutdinov, R.: Transformer-Xl: attentive language models beyond a fixed-length context. arXiv preprint arXiv:1901.02860 (2019)
22. Liu, Y., et al.: Roberta: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
23. Jo, T., Lee, J.H.: Latent keyphrase extraction using deep belief networks. Int. J. Fuzzy Logic Intell. Syst. **15**(3), 153–158 (2015)
24. Papagiannopoulou, E., Tsoumakas, G.: Local word vectors guiding keyphrase extraction. Inf. Process. Manage. **54**(6), 888–902 (2018)
25. Bennani-Smires, K., Musat, C., Hossmann, A., Baeriswyl, M., Jaggi, M.: Simple unsupervised keyphrase extraction using sentence embeddings. In: Proceedings of the 22nd Conference on Computational Natural Language Learning, Brussels, Belgium, pp. 221–229. Association for Computational Linguistics (2018)
26. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using Siamese BERT-networks. arXiv preprint arXiv:1908.10084 (2019)