CorpusRank: Corpus Information in Unsupervised Keyphrase Extraction

Isaiah Williams
Stanford University
byronw@stanford.edu

Barry Cheung
Stanford University
barryc4@stanford.edu

Abstract

Embeddings such as BERT represent syntactic information in a vector space, and are increasingly used in unsupervised keyphrase extraction. A previous paper used document-phrase similarity drawn from the embedding space of BERT and built a graph structure considering the positions of phrases in documents to rank candidate phrases by relevance. We extend their method by considering information from other documents in the corpus, and introduce a new keyphrase extraction algorithm, CorpusRank.

1 Introduction

Keyphrase extraction selects and ranks important phrases from the body of a document that encapsulate topics of the document itself (Turney, 2000). Keyphrases are used for a variety of downstream tasks such as query expansion and document classification (Papagiannopoulou and Tsoumakas, 2020).

A vast range of methods exist. Supervised keyphrase extraction algorithms can attain high accuracies, while unsupervised keyphrase extraction is more robust, domain-independent, and does not require labelled training data. Examples from the main categories of unsupervised keyphrase extraction are explored and combined in this paper: statistics-based (tf—idf), graph-based (the PageRank algorithm), embedding-based (BERT), and language model-based (n-gram) methods.

Embeddings such as GloVe (Pennington et al., 2014) and BERT (Devlin et al., 2019) represent words in a multi-dimensional embedding space. In particular, in BERT, a pre-trained model based on transformer architectures, geometry is imbued with syntactic meaning, such that "distance encodes semantic similarity, while certain directions correspond to polarities" (Coenen et al., 2019).

(Liang et al., 2021) constructs graph-based ranking for keyphrase extraction, working in the embedding space of BERT. They consider two forms of context: *global* context (document-phrase similarity), and *local* context (position of phrases in each document).

We extend their method (which we refer to as UKERank) in various directions. First, we consider the amenability of the keyphrase graph constructed to ranking using the PageRank algorithm (BERTRank). Next, we examine the effect of neighbouring documents (*corpus* context), using tf—idf, n-gram language models, and k-document centrality. Lastly, syntactic heuristics is explored as a complement to local context.

The aggregate of these models was then run on two datasets, DUC2001 and SemEval2010, and hyperparameter testing performed to identify the most suitable models for each dataset. We call our combined model CorpusRank.

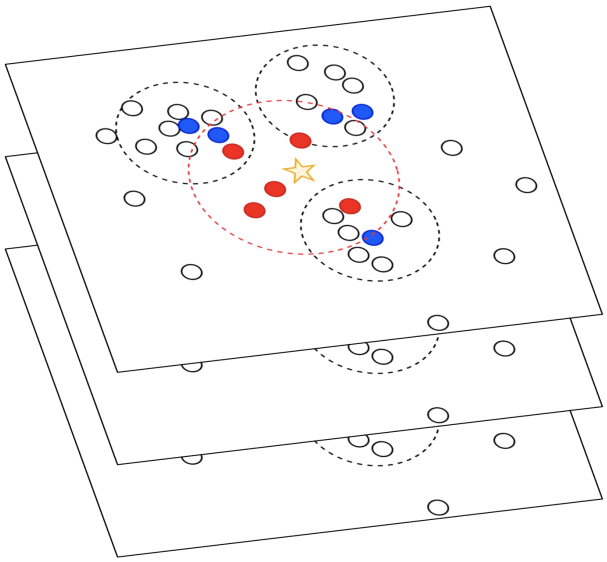

Figure 1: Tokenization converts words into vectors ○ (tokens) in an embedding space (flat sheet); the mean of these tokens is the document vector ★. Considering only *global* context (red circle) in keyphrase extraction only returns candidate phrases ● adjacent to the document vector, while considering *local* context now returns keyphrases ● representative of topics in the document (black circles). We extend this to consider adjacent documents (multiple sheets); that is, *corpus* context. Figure adapted from (Liang et al., 2021).

2 Approach

Our initial steps follow (Liang et al., 2021). We tokenize each document to produce word tokens $\{t_i\}$, and use BERT to obtain their vector representations $\{H_i\}$. We retrieve candidate phrases $\{\texttt{KP}_i\}$ containing *at least one noun possibly preceded by adjective(s)*, with vector representations $V = \{H_{\texttt{KP}_i}\}$ taken to be averages of their constituent word tokens. MaxPooling all tokens within a document produces the document vector representation H_D containing its global semantic information.

$$\begin{array}{ccccc} \{t_i\} & \xrightarrow{\texttt{BERT}} & \{H_i\} & \xrightarrow{\texttt{MaxPool}} & H_D \\ & & \downarrow \text{average} & & \\ & & V = \{H_{\texttt{KP}_i}\} & & \end{array}$$

2.1 Local and global contexts (UKERank)

(Liang et al., 2021) attributed a **global relevance** score $\mathbf{\Gamma}$ to each phrase $\texttt{KP}_i$ based on phrase-document similarity.

$$\mathbf{\Gamma}(H_{\texttt{KP}_i}) = \frac{1}{\|H_D - H_{\texttt{KP}_i}\|_1} \tag{1}$$

A graph $\mathcal{G} = (V, E)$ was constructed for each document, with vertices $V = \{H_{\texttt{KP}_i}\}$ (candidate phrases) and a set of edges $E = \{e_{ij}\}$, $e_{ij} = H_{\texttt{KP}_i} \cdot H_{\texttt{KP}_j}$ connecting each pair (i, j) of vertices. They implemented **boundary-aware centrality** to bias scores in favor of phrases appearing near the start and end of documents. For hyper-parameters (α, β, λ), define boundary function $d_b(i) = \min[i, \alpha(n-1)]$ for the i^{th} phrase, and threshold $\theta = \beta[\max(e_{ij}) - \min(e_{ij})]$. (Refer to their original paper for heuristic justifications.) Compute:

$$\mathbf{\Lambda}(H_{\texttt{KP}_i}) = \sum_{d_b(i)<d_b(j)}^{|V|} \max(e_{ij}, \theta) + \lambda \sum_{d_b(i)\geq d_b(j)}^{|V|} \max(e_{ij}, \theta) \tag{2}$$

$$\widehat{p}(\texttt{KP}_i) = e^{p(\texttt{KP}_i)} \Big/ \sum_{k=1}^{|V|} e^{p(\texttt{KP}_k)} \tag{3}$$

where $p(\texttt{KP}_i) = 1/P_i$, and P_i is the position of the first appearance of the i^{th} phrase. **Local salience** of each phrase was defined

$$\widehat{\mathbf{\Lambda}}(H_{\texttt{KP}_i}) = \widehat{p}(\texttt{KP}_i)\ \mathbf{\Lambda}(H_{\texttt{KP}_i}) \tag{4}$$

and the final score $\mathbf{S}$ of each phrase was taken to be the product of global relevance $\mathbf{\Gamma}$ and local salience $\widehat{\mathbf{\Lambda}}$. (Liang et al., 2021) used the final score to rank all candidate phrases and extract *key*phrases. We call their model UKERank.

$$\mathbf{S}(H_{\texttt{KP}_i}) = \mathbf{\Gamma}(H_{\texttt{KP}_i})\ \widehat{\mathbf{\Lambda}}(H_{\texttt{KP}_i}) \tag{5}$$

2.2 Graph-based ranking (BERTRank)

Having constructed a graph $\mathcal{G}$ of candidate phrases with edge weights e_{ij} computed from syntactic similarities in the embedding space of BERT, we posit an alternative ranking system following the famous algorithm by (Page et al., 1999). Define a similarity matrix $M_{ij} = e_{ij}$, $M \in \mathbb{R}^{|V|\times|V|}$, and take its row-wise normalized form

$$\widehat{M}_{ij} = M_{ij} \Big/ \sum_{j=1}^{|V|} M_{ij} \tag{6}$$

The vectorized **saliency score** $\vec{\xi} \in \mathbb{R}^{|V|}$ for all phrases can then be expressed as a self-consistent system,

$$\vec{\xi} = \nu \widehat{M}^{\mathrm{T}} \vec{\xi} + \frac{1-\nu}{|V|}\ \vec{\epsilon} \tag{7}$$

where ν is a damping hyper-parameter (conventionally $\nu \sim 0.85$), and $\vec{\epsilon} \in \mathbb{R}^{|V|}$ a vector of 1's. The transition matrix (in a Markovian sense)

$$\mathcal{T} = \nu \widehat{M}^{\mathrm{T}} + \frac{1-\nu}{|V|}\ (\vec{\epsilon} \otimes \vec{\epsilon}) \tag{8}$$

has a principal eigenvector $\vec{\xi}$ (of unit eigenvalue) obtained by the power method, recursively applying $\mathcal{T}$:

$$\lim_{n\to\infty} \mathcal{T}^n \Big(\vec{\epsilon}/|V|\Big) = \vec{\xi} \tag{9}$$

Saliency score for BERTRank $\mathbf{\Xi}(H_{\texttt{KP}_i}) = \xi_i$ obtained for each candidate phrase in a document is completely independent of UKERank.

2.3 Corpus context

We now consider various corpus scores. These identify keyphrases which are important in a document, but less so in other documents in the set (corpus context).

2.3.1 tf—idf

Term frequency—inverse document frequency (tf—idf) is a heuristic (Spärck-Jones, 1972) that calculates the "inverse proportion of the frequency of a word in a particular document to the percentage of documents the word appears in" (Ramos, 2003) as a measure of its relevance in a document. We use sklearn to implement these equations for word w_i in a document D_j:

$$\begin{aligned} \text{tf—idf}(w_i, D_j) &= \text{tf}(w_i, D_j) \cdot \text{idf}(w_i) \\ \text{idf}(w_i) &= \log\left(\frac{N}{\text{df}(w_i)}\right) + 1 \end{aligned} \tag{10}$$

where term frequency tf is the raw count of a word's appearance in a document, and document frequency df counts the fraction of N documents in the entire corpus containing the word w_i. The average of tf—idf scores of words $\{w_j\}$ in a candidate phrase $\texttt{KP}_i$ constitutes its corresponding corpus score, $\mathbf{K}_{\text{tf}}(H_{\texttt{KP}_i})$.

2.3.2 Pointwise Kullback—Leibler divergence

Kullback-Leibler (KL) divergence measures "the inefficiency of assuming that a distribution is q when the true distribution is p" (Cover and Thomas, 2005). It belongs to the class of f-divergences, which are information-monotonic in that the divergence measure does not increase when information is coarse-grained (Amari and Cichocki, 2010). Define KL divergence D as a sum over pointwise KL divergences δ_x:

$$D(p\|q) = \sum_x \underbrace{p(x) \log \frac{p(x)}{q(x)}}_{\delta_x(p\|q)} \tag{11}$$

Consider n-gram language models. For a sequence of n words $\mathbf{w} = w_1 w_2 \dots w_k$, the probability of a word appearing the sequence under a n-gram language model is conditioned on the previous $n-1$ words (Chen and Goodman, 1999), such that the probability of the sequence itself is

$$\begin{aligned} \mathrm{LM}^n(\mathbf{w}) &= p(w_1)\, p(w_2|w_1)\, p(w_3|w_1 w_2) \dots \\ &\quad \cdot p(w_n|w_1 w_2 \dots w_{n-1}) \\ &= \prod_{j=1}^{n} p(w_j|w_1 w_2 \dots w_{j-1}) \end{aligned} \tag{12}$$

Probabilities for bigram models can be expressed as a function of counts, c:

$$p(w_i|w_{i-1}) = c(w_{i-1} w_i) \Big/ \sum_{w_j} c(w_{j-1} w_j) \tag{13}$$

(Tomokiyo and Hurst, 2003) define two quantities, *phraseness* φ_p, which measures the loss of information under the assumption of word independence (using the unigram rather than n-gram model), and *informativeness* φ_i, the loss of information when the probability of a phrase's occurrence is considered in the corpus context c (set of all documents) rather than a single document d.

$$\begin{aligned} \varphi_p &= \delta_{\mathbf{w}}(\mathrm{LM}_d^n \| \mathrm{LM}_d^1) \\ \varphi_i &= \delta_{\mathbf{w}}(\mathrm{LM}_d^1 \| \mathrm{LM}_c^1) \end{aligned} \tag{14}$$

We combine the scores linearly to obtain KL score $\mathbf{K}_{\mathrm{kl}}$ for a candidate phrase.

$$\mathbf{K}_{\mathrm{kl}}(H_{\texttt{KP}_i}) = \varphi_p + \varphi_i \tag{15}$$

2.3.3 *k*-document centrality

We may alternatively limit our corpus context to the k semantically nearest documents, as measured by:

$$S_c(H_D^{(i)}, H_D^{(j)}) = \frac{H_D^{(i)} \cdot H_D^{(j)}}{\|H_D^{(i)}\|_2 \, \|H_D^{(j)}\|_2} \tag{16}$$

(Wan and Xiao, 2008) considered a group of nearest documents in their ranking algorithm, ExpandRank. We apply this to modify the earlier measure of local salience (Liang et al., 2021) to sum over k nearest documents, weighted by cosine similarity.

$$\begin{aligned} \mathbf{\Lambda}_k(H_{\texttt{KP}_i}) &= \sum_{l=0}^{k} S_c(H_D^{(0)}, H_D^{(l)}) \sum_{d_b(i)<d_b(j)}^{|V|} \max(e_{ij}, \theta) \\ &\quad + \lambda \sum_{l=0}^{k} S_c(H_D^{(0)}, H_D^{(l)}) \sum_{d_b(i)\geq d_b(j)}^{|V|} \max(e_{ij}, \theta) \end{aligned} \tag{17}$$

In an analogous manner, a final score $\mathbf{S}_k$ was assigned to each candidate phrase $\texttt{KP}_i$, now accounting for global, local, and k-document contexts.

$$\mathbf{S}_k(H_{\texttt{KP}_i}) = \mathbf{\Gamma}(H_{\texttt{KP}_i})\, \widehat{p}(\texttt{KP}_i)\, \mathbf{\Lambda}_k(H_{\texttt{KP}_i}) \tag{18}$$

For $k = 0$, this reduces to the score $\mathbf{S}$ of `UKERank` (Liang et al., 2021).

The new embeddings resulted in slightly poorer performance on each of the three metrics. This result seems to disprove our hypothesis that improved performance on Masked Language Modeling would also result in improved token embeddings. Given that these token embeddings were used for phrase and document embeddings in particular, the disconnect between training task and use-case likely caused our model to be weaker.

2.4 Syntactic heuristics

2.4.1 Score

In the process of determining which methods would allow us to select for the most representative keyphrases, we started off with multiple hypotheses about which linguistic features are most highly correlated with the best keyphrases of a document. We refer to (Barker and Cornacchia, 2000) for their use of noun phrase heads to extract keyphrases as the basis for using linguistic features in unsupervised keyphrase extraction. Of our several hypotheses, we settled on testing three in particular, and theorized that candidate phrases which are

1. the noun phrase complement of a preposition, or
2. the subject noun phrase of a sentence, or
3. the object noun phrase of a sentence

would more closely correlate to golden keyphrases.

We represented these binary linguistic features in our model by first gathering three separate lists of candidate

phrases that met each of the respective criteria above.

We calculate a **syntax score** $\mathbf{\Sigma}_\varphi$ for each feature φ

$$\mathbf{\Sigma}_\varphi(H_{\text{KP}_i}) = 1 + \beta p_\varphi \tag{19}$$

where $p_\varphi \in \{0, 1\}$ is determined by whether the candidate phrase meets one of the three criteria: $\varphi \in$ {prepcomp, subject, object}.

2.4.2 Preliminary Findings

In preliminary experiments, we calculated a separate syntax score for each feature rather than aggregating each feature into our final syntax. We found that only the score

$$\mathbf{\Pi}(H_{\text{KP}_i}) \equiv \mathbf{\Sigma}_{\text{prepcomp}}(H_{\text{KP}_i}) = 1+\beta p_{\text{prepcomp}} \tag{20}$$

contributed positively to our results while the scores representing other linguistic features had a negligible or negative impact on our results.

In our final experiments, we only evaluated whether a candidate phrase was the complement of a preposition to the exclusion of other features. Therefore, we declare the score listed above to be the **preposition score**, $\mathbf{\Pi}$ and we only use this score in the evaluation of our final results.

2.5 Evaluation metrics

We use the evaluation metrics highlighted in (Liang et al., 2021), following the common practice of evaluating performance in terms of f-measure at the top N keyphrases (F1@N), with stemming applied to candidate phrases as well as the golden key phrases. We report F1@5, F1@10, and F1@15 for each of the datasets that we use. (Naseem et al., 2021)

3 Dataset

We evaluate our model against two datasets, DUC2001 and SemEval2010. DUC2001 contains 308 long-length news articles with their corresponding gold-truth key phrases, which were manually labeled as the most significant adjective/noun phrases within the document. SemEval2010 is a similarly labeled data-set containing 500 full-length ACM papers and their respective labeled keyphrases. These two data-sets provide a strong range of testing capacity due to their diversity of topics. Additionally, while news articles tend to have more informal language closer to natural speech, ACM papers are more formal and introduce novel topics into the general vocabulary. These two complementing datasets provide a robust approximation to the true text population. It is important to note that the average length of texts in DUC2001 and SemEval2010 are 847 and 1588 words respectively.

While SemEval2010 proves to be the more difficult of the two benchmarks, given its lower F1 Score at the previous state of the art. We find that that the percentage of improvement on SemEval2010 is notably higher than the percentage of improvement on DUC2010. This leads us to believe that our model performs better the longer a given document. Improved performance on longer documents is consistent with our assumptions given our robust modeling of candidate phrase position as well as position relative to other candidate phrases. With more information, our model has stronger predictive capabilities.

4 Neural networks and language models

The strength of our model is highly correlated to the strength of our token embeddings. Candidate phrase embeddings are determined by averaging its corresponding token embeddings. Document embeddings are created by Maxpooling the phrase embeddings H_i. Therefore, we attempted to improve the BERT model by continued training of BERT on domain-specific data. (Gururangan et al., 2020) showed that continued training on domain specific data showed improved accuracy on Masked Language Modeling tasks relevant to that domain. Since the BERT model previously in use also was pre-trained via the Masked Language Modeling Task, our hypothesis was that this improved accuracy would extend to token embeddings.

We initialized our model using the bert-base-uncased (Devlin et al., 2019) model. The model was trained using approximately 18,000 sentences from the CC-NEWS (Hamborg et al., 2017) dataset that was scraped via the news-please crawler. This dataset contains 708241 English language news articles published between Jan 2017 and December 2019. To create our training set, we masked 15 percent of the tokens contained in each of our 18,000 sentences. We then trained BERT on dataset with a batch size of 4 for 4 epochs. Embeddings were obtained using the same conventions as in (Liang et al., 2021). The candidate phrase embeddings were taken from the average of its token embeddings, and the document embeddings were obtained by MaxPooling its corresponding phrase embeddings.

4.1 Domain-Specific Results

Since this model was trained specifically on a news related domain, we calculate the F1-score at 5, 10, and 15 to compare to that of the off-the-shelf BERT model. The scores can be found in the Results section.

Models	DUC2001			SemEval2010		
	F1@5	F1@10	F1@15	F1@5	F1@10	F1@15
TF-IDF	9.21	10.63	11.06	2.81	3.48	3.91
YAKE	12.27	14.37	14.76	11.76	14.4	15.19
TextRank	11.80	18.28	20.22	3.80	5.38	7.65
UKERank	28.62	35.52	36.29	13.02	19.35	21.72
Our Model	**33.10**	**38.88**	**39.97**	**17.40**	**22.60**	**25.98**

Table 1: Comparison of our model with other baselines.

Models	DUC2001			SemEval2010		
	F1@5	F1@10	F1@15	F1@5	F1@10	F1@15
UKERank	28.62	35.52	36.29	13.02	19.35	21.72
Our Model	33.10	38.88	39.97	17.40	22.60	25.98
Our Model - DUC	**34.80**	**40.24**	**40.86**	14.17	21.56	25.04
Our Model - SemEval	28.73	35.94	38.17	**20.26**	**25.25**	**27.60**

Table 2: Our model fine-tuned to each Dataset
DUC: Hyperparameters are the same, k = 20
SemEval: k = 1, Hyperparameters = [.75, .2, .03, .95, 1]

5 Results

5.1 Combined Models

To evaluate our model, we ranked all of our candidate keyphrases using the total score as shown previously. Surprisingly, we found that $k = 1$ for k-document centrality provided the best generalization between the datasets. Given the joint use of multiple heuristics and models our final ranking score $\mathbf{T}$ is tuned by 5 hyper-parameters $[a, b, c, d, e]$.

$$\mathbf{T}(H_{\mathtt{KP}_i}) = \widehat{p}(\mathtt{KP}_i) \cdot \mathbf{K}^2_{\mathbf{kl}}(H_{\mathtt{KP}_i}) \cdot \mathbf{K}^b_{\mathbf{tf}}(H_{\mathtt{KP}_i} \big(1 + c\mathbf{\Pi}(H_{\mathtt{KP}_i})\big) \cdot \mathbf{\Xi}^d(H_{\mathtt{KP}_i}) \cdot \mathbf{\Sigma}^e_k(H_{\mathtt{KP}_i}) \tag{21}$$

To determine these hyper-parameters, we took 10,000 samples, randomly selecting a value between 0 and 1 for each value with one exception. For hyper parameter c, we selected random values between -0.1 and $-.1$. Our intuition here was that negative hyper parameter c could model a negative correlation between gold truth keyphrases and complements of prepositions. As a result, we found the best values for the list of hyper parameters to be [0.1, 0.2, 0.06, 0.9, 0.7]. In spite of the robust syntactical and thematic difference between the two corpuses, these values gave strong scores on both DUC2001 and SemEval2010 proving the models potential at generalization. These hyper parameters represent heavy dependency on the BERT Rank and the original UKERank model.

5.2 Result Metrics and Domain-tuned Experiments

As you can see in Table 1, our model outperforms the previous state of the art benchmarks in all metrics on both DUC2001 and SemEval2010 making our model CorpusRank the current- state of the art. In development, we found that just applying the TF-IDF score to the original UKERank model gave already significant improvements over the previous state of the art. This version of the model runs very quickly and still provides results close to the state of the art.

We also found that our combination of models performs exceptionally well if the hyper parameters are fine-tuned to the individual datasets. We can see the results of this in Table 2. We found that when k is increased to 20 we get meaningful performance boost on DUC2001. This means a node graph is created with edges between every candidate phrase in a given document and every candidate phrase in its k=20 nearest documents. Such results are consistent with intuition given that the corpus represents news articles and news articles often reveal inner importance relative to one another.

On SemEval2010, the Kl divergence model and BERTRank model played strong roles, to greatly increase the F1@5 score. We see over a one third increase in the score. Surprisingly, high k values greatly hurt the score as can be noted on CorpusRank-DUC. Given that SemEval2010 is a corpus of scientific papers with an average length of 1600 words, the k-document centrality method seems to hurt the most prominent keyphrases Whereas the surprise factor of a keyphrase along with its internal relationship in the single document weight its ranking. This can be shown by the new hyperparameters for this model $[.75, .2, .03, .95, 1]$.

Our model can be generalized using the primary hyper parameters reported to tackle a general domain space of documents. It can also be fine tuned to a dataset to see great performance boost depending on the structure of the corpus.

6 Future work

In this paper, we utilize BERT embeddings. The BERT embeddings are used to model similarity between two candidate phrases, a candidate phrase and document, and two documents. Given its recurrent use in our model, further development on this embedding space will greatly improve its strength. In future work, we will focus on creating more specialized representations for documents and candidate phrases. One avenue to achieve this would be to fine-tune a BERT model on a key-phrase specific task or to train even larger domain-specific models.

References

Shun-ichi Amari and Andrzej Cichocki. 2010. Information geometry of divergence functions. *Bulletin of the Polish Academy of Sciences*, 58.

Ken Barker and Nadia Cornacchia. 2000. Using noun phrase heads to extract document keyphrases. In *Advances in Artificial Intelligence*, pages 40–52, Berlin, Heidelberg. Springer Berlin Heidelberg.

Stanley F. Chen and Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech Language*, 13(4):359–394.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda B. Viégas, and Martin Wattenberg. 2019. Visualizing and measuring the geometry of BERT. *CoRR*, abs/1906.02715.

Thomas Cover and Joy Thomas. 2005. *Elements of Information Theory*. John Wiley Sons, Ltd.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. pages 4171–4186. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks.

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: A generic news crawler and extractor. In *Proceedings of the 15th International Symposium of Information Science*, pages 218–223.

Xinnian Liang, Shuangzhi Wu, Mu Li, and Zhoujun Li. 2021. Unsupervised keyphrase extraction by jointly modeling local and global context. pages 155–164. Association for Computational Linguistics.

Usman Naseem, Imran Razzak, Shah Khalid Khan, and Mukesh Prasad. 2021. A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 20(5).

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66.

Eirini Papagiannopoulou and Grigorios Tsoumakas. 2020. A review of keyphrase extraction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Juan Ramos. 2003. Using tf-idf to determine word relevance in document queries.

Karen Spärck-Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. page 33–40. Association for Computational Linguistics.

Peter D. Turney. 2000. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2:303–336.

Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence — Volume 2*, page 855–860. AAAI Press.