EntropyRank: Unsupervised Keyphrase Extraction via Side-Information Optimization for Language Model-based Text Compression

Alexander Tsvetkov [1 2] **Alon Kipnis** [1]

Abstract

We propose an unsupervised method to extract keywords and keyphrases from texts based on a pre-trained language model (LM) and Shannon's information maximization. Specifically, our method extracts phrases having the highest conditional entropy under the LM. The resulting set of keyphrases turns out to solve a relevant information-theoretic problem: if provided as side information, it leads to the expected minimal binary code length in compressing the text using the LM and an entropy encoder. Alternately, the resulting set is an approximation via a causal LM to the set of phrases that minimize the entropy of the text when conditioned upon it. Empirically, the method provides results comparable to the most commonly used methods in various keyphrase extraction benchmark challenges.

1. Introduction

1.1. Motivation

Keyphrase extraction can be described as an information distillation process of a document into a series of words. These words are later used as a proxy for document representation, to be utilized in various downstream tasks such as extractive summarization, information retrieval, clustering, document categorization, and query expansion (Hasan & Ng, 2014; Medelyan & Witten, 2008). Although the problem involves a task that is native to information theory, i.e., extracting information subject to a constraint, to the best of our knowledge, non of the existing methods directly optimize Shannon's information or strive to solve an information transmission problem. The purpose of the current work is to suggest a method that stems from such information-theoretic principles and to demonstrate that it performs comparable to the state-of-the-art methods.

1.2. Contribution

We present a novel method of unsupervised keyphrase extraction denoted EntropyRank that strives to minimize Shannon's entropy of text given the keyphrases. The conditional entropy is evaluated with respect to a pre-trained LM, typically a large LM based on transformer deep neural networks (Vaswani et al., 2017). The resulting set of keyphrases has a relevant operational interpretation: the set that provides the maximal reduction in the expected binary code length (bits) when compressing the text using the LM and an entropy encoder, while the keyphrases and their locations are provided as side information. Works utilizing this form of compression but without side information are known to achieve state-of-the-art results on lossless text compression (Izacard et al., 2019; Mahoney, 2023). The extraction principle of EntropyRank is reminiscent of lossless compression methods that provide the most difficult parts to predict as side information (Caire et al., 2003).

While our method is derived directly from information theoretic principles, it appears to perform well empirically, attaining results comparable to the most commonly used method over a series of benchmark tasks; see the report in Section 4.

1.3. Background

Keyphrase extraction can be supervised or unsupervised, with the former requiring labeled training data and the latter being more domain-independent (Sahrawat et al., 2020). In many situations, manual labeling is impractical or unavailable due to domain adaptation challenges hence unsupervised keyphrase extraction is the only viable option.

Unsupervised keyphrase extraction methods can be categorized into four groups based on the features they use to rank candidates: statistical, embedding-based, graph-based, and generative. Statistical methods such as RAKE and YAKE use a pre-trained model of combining features such as TF-IDF, relative position, and co-occurrence (Cam-

[*] Equal contribution [1] School of Computer Science, Reichman University, Herzliya, Israel [2] Microsoft, Herzliya, Israel. Correspondence to: Alexander Tsvetkov <alexander.tsvetkov@post.runi.ac.il>, Alon Kipnis <alon.kipnis@runi.ac.il>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

pos et al., 2018). Embedding-based methods such as KeyBert and PatternRank use word or sentence embeddings to measure the relevance of text chunks to the document (Bennani-Smires et al., 2018; Grootendorst, 2020; Schopf et al., 2022). Graph-based methods such as TextRank construct a co-occurrence graph of words or phrases and apply centrality measures to score them (Mihalcea & Tarau, 2004; Rose et al., 2010). Some methods also combine textual or semantic features with graph or embedding features to form hybrid models(Mahata et al., 2018a;b). The newest generative methods use instruction-tuned LMs (Ouyang et al., 2022) directly with a prefixed context of the text and an instruction in the form of a contextualized prompt to generate keywords directly from the text. However, most of these approaches have some inherent drawbacks. Specifically, statistical and graph-based methods rely on local corpus features, hence they disregard natural language regularities and typically require some hyper-parameter tuning. Semantic and embedding methods tend to struggle with phrases that do not match the document's context without proper tuning. Generative models, on the other hand, can produce unreliable and unpredictable results due to biases and hallucinations (Ji et al., 2023), which hinder their use in practice. In contrast, EntropyRank incorporates language regularities and semantics directly from the LM, suggesting that it works well whenever the LM reasonably predicts tokens under a cross-entropy (log) loss, the typical training objective of modern LMs.

1.4. Organization

The rest of this paper is organized as follows. In Section 2 we describe the method. In Section 3 we analyze the method under the lens of source coding in information theory. In Section 4 we report on empirical results. Concluding remarks are provided in Section 5.

2. Method Description

Let P_{model} be a causal LM, i.e., a set of conditional probability distributions over sequences of tokens $w_{1:n} = (w_1, \ldots, w_n)$ of the form

$$P_{\mathsf{model}}(\cdot|w_{1:i-1}) = \Pr[W_i|W_{1:i-1} = w_{1:i-1}], \quad i \leq n.$$

Above and throughout we use the notation $u_{1:0} := \emptyset$ for any sequence u. By extension, P_{model} also provides conditional probabilities of the form $\Pr[X_i|X_{1:i-1} = x_{1:i-1}]$ where $x_{1:n} = (x_1, \ldots, x_n)$ is a sequence of text phrases and each phrase is a sequence of tokens. Specifically, if phrase x_i consists of tokens $(w_{i,1}, \ldots, w_{i,n_i})$, then the probability of x_i is

$$P_{\mathsf{model}}(x_i|x_{1:i-1}) = \prod_{j=1}^{n_i} P_{\mathsf{model}}(w_{i,j}|x_{1:i-1}, w_{i,1:j-1}).$$

Algorithm 1 EntropyRank

```
    **Input:** Text document $D$, number of keyphrases $k$, language model $P_{\mathsf{model}}$
    Segment text to phrases $x_{1:n} = (x_1, \ldots, x_n)$
    **for** $i = 1$ **to** $n$ **do**
        $H_i \leftarrow H(P_{\mathsf{model}}(\cdot|x_{1:i-1}))$
    **end for**
    $J^* = \arg\max_{J:|J|\leq k} \sum_{j\in J} H_i$
    **return** $\{X_j\}_{j\in J^*}$
```

Our method first segments the document into phrases, for example using noun phrases or stop words as in (Schopf et al., 2022; Rose et al., 2010). Given the segmented document $x_{1:n} = (x_1, \ldots, x_n)$, we refer to

$$H_i := H(X_i|X_{1:i-1} = x_{1:i-1}) = H(P_{\mathsf{model}}(\cdot|x_{1:i-1})), \tag{1}$$

as the entropy of the i-th phrase under the LM. Here $P_{\mathsf{model}}(\cdot|x_{1:i-1})$ is the distribution of the i-th phrase in the document given the previous $i-1$ phrases as provided by the LM and H_i is Shannon's entropy of this distribution (Cover & Thomas, 2012). Note that H_i is a function of the phrases preceding x_i but not of x_i itself.

Our method outputs a set of phrases $\{X_j,\, j \in J^*\}$, $J^* \subset \{1, \ldots, n\}$, that maximizes the sum of phrase entropies subject to the cardinality constraint of at most k elements. Namely, J^* maximizes

$$\sum_{j\in J} H_j, \quad \text{subject to} \quad |J^*| \leq k; \tag{2}$$

the entire procedure is summarized in Algorithm 1. Another useful practice is to report the smallest set of keyphrases such that the sum in (2) exceeds some specified bit threshold. An operational interpretation of the phrase entropies and this bit threshold is given next.

3. Information Theoretic Analysis

We provide two viewpoints to motivate EntropyRank.

3.1. Lossless text compression with side information

Consider the problem of compressing the text using a binary code when a set of phrases indexed by J is provided as side information while the cardinality of J is restricted. We can interpret the entropy H_i of (1) as the amount of information keyphrase x_i provides on the text in the following sense. It is the expected reduction in the number of bits needed to encode the text using the LM P_{model} and an entropy encoder when x_i and its location i are provided as side information (regardless of the distribution of the text).

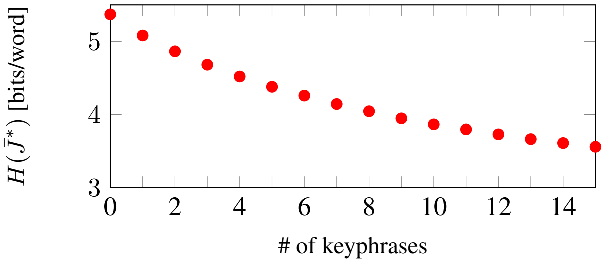

Figure 1. Expected remaining normalized text entropy $H(\bar{J}^*)$ versus the number of keyphrases. The remaining entropy is the expected number of bits needed to encode the text via a LM and an entropy encoder when the keyphrases are provided as side information.

To better explain what we mean by this kind of encoding, suppose that we encode $x_{1:n}$ when (x_m, m), $2 < m \le n$ is provided as side information and using an arithmetic encoder as the entropy encoder (Langdon, 1984). Starting with x_1, we find a partition of the interval $[a_0, b_0) = [0, 1)$ according to the distribution $P_{\mathsf{model}}(\cdot|\emptyset)$ in a pre-determined order; we denote by $[a_1, b_1)$ the interval corresponding to x_1 in this partition. Next, we partition $[a_1, b_1)$ according to $P_{\mathsf{model}}(\cdot|x_1)$ in the same order and denote by $[a_2, b_2)$ the interval corresponding to x_2 in this partition. The situation continues until we reach x_m, in which case we add x_m as a context to the LM but otherwise ignore it and move to partition $[a_{m-1}, b_{m-1})$ according to $P_{\mathsf{model}}(\cdot|x_{1:m})$. The resulting encoded representation of the text is the shortest binary representation falling within the interval corresponding to x_n at the last step with the leading zero removed. This encoding process is clearly reversible given P_{model} and (x_m, m). The extension to more than one phrase provided as side information is straightforward. This form of encoding but without incorporating side information is used in (Izacard et al., 2019; Liu et al., 2019; Goyal et al., 2021). Our method ranks each phrase according to the entropy H_i of (1) associated with its location in the text. By design, the highest-ranked keyphrase provides more information (in the sense of expected code length reduction) on the text than the second-highest, and so forth. The expected code length is the remaining text entropy provided by the sum $H(\bar{J}^*) := \sum_{j \notin J^*} H_i$. This can be seen empirically in Figure 1, showing the average entropy of keyphrases based on their rank over the Inspec dataset (Hulth, 2003).

3.2. Approximating the Information Maximizing Set

EntropyRank also arises as a tractable approximation to an optimal information-theoretic solution of the keyphrases extraction problem. In order to formulate this problem, denote by $X_{1:n} = (X_1, \ldots, X_n)$ the sequence of phrases constituting a document, viewed as random variables over a dictionary. Let $J \subset \{1, \ldots, n\}$ be a set of indexes of phrases in this document. We seek a set of keyphrases indexed by $J^\dagger$ that captures most of the information as measured by Shannon's entropy $H(X_{1:n})$. Namely, with $\bar{J} = \{1, \ldots, n\} \setminus J$, $J^\dagger$ minimizes $H(X_{\bar{J}}|X_J)$ in the decomposition

$$H(X_{1:n}) = H(X_J) + H(X_{\bar{J}}|X_J) \tag{3}$$

subject to the cardinality constraint $|J| \le k$. Since the mutual information decomposes as

$$\begin{aligned} I(X_{1:n}; X_J) &= H(X_{1:n}) - H(X_{1:n}|X_J) \\ &= H(X_{1:n}) - H(X_{\bar{J}}|X_J), \end{aligned}$$

we can also think of $J^\dagger$ as the maximizer of the mutual information between the set of keyphrases and the entire text subject to the cardinality constraint.

It is usually intractable to minimize $H(X_{\bar{J}}|X_J)$ directly and evaluate $J^\dagger$ for large texts due to the large search space and the need to evaluate non-causal conditional probability expressions[1]. We turn to seek an approximation to $J^\dagger$. We decompose the entropy of the text as

$$H(X_{1:n}) = \sum_{i \in J} H(X_i|X_{1:i-1}) + \sum_{i \in \bar{J}} H(X_i|X_{1:i-1}) \tag{4}$$

and look for a set J^* that maximizes $\sum_{i \in J} H_i$, i.e. the observed version of the first sum in (4), subject to the cardinality constraint. Our method provides the set J^* under the assumption that the conditional distribution of the text is provided by the LM. In the case of a distributional mismatch between the LM and the text, an analogous logic applies when replacing entropy with cross-entropy (Cover & Thomas, 2012).

4. Empirical Results

4.1. Implementation

We use GPT-Neo 1.7B (Black et al., 2021), a pre trained LM trained on the PILE dataset (Gao et al., 2020), to estimate the natural language distribution of the text. We segment the text into noun phrases that match the parts of speech tag patterns <J.*>*<N.*>+, capturing zero or more adjectives followed by one or more nouns. We rank these segments by the sum of the entropy of their words and extract the top k candidates with the highest entropy as keyphrases.

4.2. Datasets

We evaluated our method on three common datasets with expert annotations, which are often used to evaluate keyphrase extraction methods in the literature.

[1] We only have an interface to evaluate causal conditional probabilities of the form $\Pr[X_i|X_{1:i-1} = x_{1:i-1}]$, hence we must marginalize over the dictionary to evaluate non-causal probabilities like $\Pr[X_2|X_1 = x_1, X_3 = x_3]$. Each marginalization involves several thousands of LM evaluations hence the entire search quickly becomes impractical.

	METHOD	@5 KEYPHRASES				@10 KEYPHRASES			
		P	R	F1	RO1	P	R	F1	RO1
INSPEC	PatternRank	**32.9**	**30.99**	**29.42**	**44.51**	**28.5**	**49.7**	**33.85**	**48.71**
	EntropyRank	32.21	29.18	28.26	43.8	27.47	47.12	32.39	48.15
	RAKE	21.34	20.6	19.32	37.39	22.24	39.71	26.63	43.29
	YAKE	17.33	17.02	15.74	33.3	14.34	27.4	17.46	30.04
	TextRank	30.45	27.83	26.86	39.61	25.51	44.52	30.24	43.8
SE 2010	PatternRank	7.95	4.73	5.75	**23.41**	6.8	7.83	7.06	**21.82**
	EntropyRank	4.92	2.56	3.31	15.31	5.53	5.93	5.58	18.98
	RAKE	0.08	0.05	0.06	5.18	0.04	0.05	0.04	9.17
	YAKE	**11.72**	**6.31**	**7.98**	16.97	**10.45**	**11.06**	**10.46**	20.59
	TextRank	4.84	2.63	3.3	14.88	4.1	4.44	4.12	14.81
SE 2017	PatternRank	**35.52**	**16.24**	**21.43**	**29.08**	**32.04**	**28.5**	**28.87**	**42.9**
	EntropyRank	28.36	12.88	17.0	26.15	25.82	23.09	23.3	39.72
	RAKE	18.0	8.52	11.12	25.01	20.64	18.78	18.89	39.53
	YAKE	18.16	8.2	10.84	19.85	17.86	16.12	16.17	28.58
	TextRank	25.68	11.73	15.48	25.6	24.2	21.59	21.88	37.59

Table 1. Performance evaluation of keyphrase extraction models on benchmark datasets. **Bolded** values indicate highest score, underlined values indicate our method.

Inspec (Hulth, 2003)- abstracts of 2,000 English scientific papers from the Inspec database.

SE-2010 (Kim et al., 2010)- full scientific articles that are obtained from the ACM Digital Library.

SE-2017 (Augenstein et al., 2017)- abstracts of 500 English scientific papers from the ScienceDirect publications.

4.3. Baseline Methods

We compared our method to popular baseline methods:

PatternRank (Schopf et al., 2022) - an extension of KeyBERT which extracts the noun phrases with the highest document similarity.

RAKE (Rose et al., 2010) - extracts phrases based on delimiters(stopwords, punctuation) and co-occurrences scoring.

YAKE (Campos et al., 2018) - based on statistical features such as term frequency, and position.

TextRank (Mihalcea & Tarau, 2004) - applies a graph-based ranking algorithm to words and phrases.

4.4. Evaluation Metrics

To assess the quality of our key phrase extraction method, we used classification and summarization metrics. The former included recall, precision, and f1 scores at different k values, measuring the agreement with the ground truth labels. The latter was ROUGE1(Lin, 2004), which calculates the single word overlap between the concatenated key phrases and the gold key phrases, reflecting the information distillation aspect of the task.

4.5. Discussion

The benchmark results on Table 1 show that EntropyRank performs well on short text datasets, such as SE2017 and INSPEC, where it achieves similar results to PatternRank and surpasses all the other methods. However, it struggles with long texts, such as SE2010, possibly due to the low rank our method gives to phrases with many occurrences which in long texts are more likely to be keyphrases. This limitation is easy to resolve in practice by using a simple term frequency-based extractor in parallel to EntropyRank. A comparison of the extracted key phrases by EntropyRank and PatternRank on INSPEC reveals a low Jacard similarity score of 0.21, indicating that they produce different and complementary results. Thus, EntropyRank has shown to be a suitable keyphrase extraction method for short texts and can enhance other methods as a complementary approach.

5. Conclusions

We presented EntropyRank, a novel unsupervised method for keyphrase extraction based on the information-theoretic principles of conditional entropy minimization under a pretrained language model. The method is simple and very direct to apply. Nevertheless, empirical results demonstrate that our method is comparable to state-of-the-art methods on several benchmark challenges.

In future work, we plan to explore the connection between our method and task-oriented lossy compression. For example, by evaluating the impact of our keyphrases on downstream tasks such as IR, clustering, or categorization.

References

Augenstein, I., Das, M., Riedel, S., Vikraman, L., and McCallum, A. Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. *arXiv preprint arXiv:1704.02853*, 2017.

Bennani-Smires, K., Musat, C., Hossmann, A., Baeriswyl, M., and Jaggi, M. Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pp. 221–229, 2018.

Black, S., Gao, L., Wang, P., Leahy, C., and Biderman, S. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow, March 2021. URL https://doi.org/10.5281/zenodo.5297715. If you use this software, please cite it using these metadata.

Caire, G., Shamai, S., and Verdu, S. Lossless data compression with error correcting codes. In *IEEE International Symposium on Information Theory, 2003. Proceedings.*, pp. 22. IEEE, 2003.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A. M., Nunes, C., and Jatowt, A. Yake! collection-independent automatic keyword extractor. In *Advances in Information Retrieval: 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings 40*, pp. 806–810. Springer, 2018.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. John Wiley & Sons, 2012.

Gao, L., Biderman, S., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

Goyal, M., Tatwawadi, K., Chandak, S., and Ochoa, I. Dzip: Improved general-purpose loss less compression based on novel neural network modeling. In *2021 Data Compression Conference (DCC)*, pp. 153–162. IEEE, 2021.

Grootendorst, M. Keybert: Minimal keyword extraction with bert., 2020. URL https://doi.org/10.5281/zenodo.4461265.

Hasan, K. S. and Ng, V. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1262–1273, 2014.

Hulth, A. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp. 216–223, 2003.

Izacard, G., Joulin, A., and Grave, E. Lossless data compression with transformer. 2019. URL https://bellard.org/nncp/nncp.pdf.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., and Fung, P. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023.

Kim, S. N., Medelyan, O., Kan, M.-Y., and Baldwin, T. SemEval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 21–26, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL https://aclanthology.org/S10-1004.

Langdon, G. G. An introduction to arithmetic coding. *IBM Journal of Research and Development*, 28(2):135–149, 1984.

Lin, C.-Y. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

Liu, Q., Xu, Y., and Li, Z. Decmac: A deep context model for high efficiency arithmetic coding. In *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 438–443. IEEE, 2019.

Mahata, D., Kuriakose, J., Shah, R., and Zimmermann, R. Key2vec: Automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 634–639, 2018a.

Mahata, D., Shah, R. R., Kuriakose, J., Zimmermann, R., and Talburt, J. R. Theme-weighted ranking of keywords from text documents using phrase embeddings. In *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, pp. 184–189. IEEE, 2018b.

Mahoney, M. Large text compression benchmark, 2023. URL http://www.mattmahoney.net/dc/text.html.

Medelyan, O. and Witten, I. H. Domain-independent automatic keyphrase indexing with small training sets. *Journal of the American Society for Information Science and Technology*, 59(7):1026–1040, 2008.

Mihalcea, R. and Tarau, P. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404–411, 2004.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Rose, S., Engel, D., Cramer, N., and Cowley, W. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, pp. 1–20, 2010.

Sahrawat, D., Mahata, D., Zhang, H., Kulkarni, M., Sharma, A., Gosangi, R., Stent, A., Kumar, Y., Shah, R. R., and Zimmermann, R. Keyphrase extraction as sequence labeling using contextualized embeddings. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42*, pp. 328–335. Springer, 2020.

Schopf, T., Klimek, S., and Matthes, F. Patternrank: leveraging pretrained language models and part of speech for unsupervised keyphrase extraction. *arXiv preprint arXiv:2210.05245*, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.