Keyphrase Extraction Using BERT Embeddings

Prepared by: Abdullah Al-Dhayan, Saad Al-Subaie, and Abdulelah Al-Haidari
Supervised by: Dr. Mohsin Bilal

---

1. Introduction

Keyphrase extraction is a critical task in natural language processing (NLP) that enables efficient summarization of documents by identifying key topics. These keyphrases facilitate text indexing, search optimization, and document classification. Traditional approaches often rely on statistical methods, such as Term Frequency-Inverse Document Frequency (TF-IDF), or rule-based algorithms, which, while effective, fall short in capturing deeper semantic meanings and contextual relationships.

With the advent of deep learning and transformer models, like BERT (Bidirectional Encoder Representations from Transformers), keyphrase extraction has entered a new era. BERT-based embeddings provide a robust mechanism to capture bidirectional contextual understanding, enabling the extraction of both present and absent keyphrases from documents. This capability ensures not only higher precision but also better adaptability to diverse text types and domains, particularly in specialized fields such as scientific literature.

This paper explores the implementation and effectiveness of BERT embeddings in keyphrase extraction, focusing on leveraging SciBERT, a variant of BERT trained on scientific texts. By comparing state-of-the-art methodologies and datasets, we aim to highlight the transformative impact of embedding-based approaches on this crucial NLP task.

---

2. Related Work

2.1 Deep Keyphrase Generation

Meng et al. (2017) introduced a groundbreaking approach to keyphrase extraction using a sequence-to-sequence (Seq2Seq) model. The study tackled the limitations of traditional methods by incorporating an encoder-decoder architecture with attention and copy mechanisms. This enabled the model to generate both present and absent keyphrases, addressing a major challenge in the field.

The key contributions of this work include:

1. Generating semantically relevant keyphrases even if they are not explicitly mentioned in the text.
2. Integrating attention and copy mechanisms to improve focus on relevant parts of the text.

However, the model exhibited certain limitations, such as favoring shorter keyphrases and requiring substantial computational resources.

2.2 Keyphrase Extraction Using Deep Recurrent Neural Networks on Twitter

Zhang et al. (2016) explored keyphrase extraction from short texts, such as tweets, using a deep recurrent neural network (RNN). The proposed model incorporated a two-layer architecture: the first

layer focused on keyword identification, while the second layer captured keyphrase boundaries. By combining keyword ranking and keyphrase generation in a unified process, the model demonstrated significant improvements over traditional approaches.

Key takeaways from this work include:

1. Unified processing of keyword ranking and keyphrase extraction reduces error propagation.
2. The model adapts well to short, noisy text environments like social media platforms.

Despite its success, the model faced challenges with limited linguistic features in short texts and required a carefully curated dataset for effective training.

---

3. Dataset: KP20k Overview

The KP20k dataset is a benchmark dataset widely used for keyphrase extraction research. It comprises over 20,000 scientific paper abstracts annotated with manually labeled keyphrases.

3.1 Dataset Details

- **Fields:**
    - **ID:** A unique identifier for each document.
    - **Document:** Scientific paper abstracts.
    - **Keyphrases:** Manually annotated keyphrases, including both present and absent keyphrases.
- **Purpose:**
  The dataset is designed for evaluating keyphrase extraction models, offering a rich and diverse set of scientific content for training and benchmarking.

---

4. **Methodology and Embedding Effectiveness**

This section details the methodology adopted for keyphrase extraction using SciBERT, highlighting the role of dynamic embeddings and advanced training strategies.

4.1 Methodology

The methodology leverages the power of SciBERT, a BERT variant specifically pre-trained on scientific texts, to enhance keyphrase extraction performance. Below are the steps taken:

4.2. Data Preparation:

- Dataset:

The KP20k dataset, containing scientific paper abstracts and annotated keyphrases, was chosen for its diversity and quality. A subset of 5000 examples was selected for training to manage computational resources effectively.

- Text Cleaning:

Text was pre-processed to remove non-alphanumeric characters and tokenize into words. A custom cleaning function ensured compatibility with the SciBERT tokenizer.

- BIO Tagging:

  - Keyphrases were labelled using the BIO schema:

    - B: Indicates the beginning of a keyphrase.

    - I: Marks continuation within a keyphrase.

    - O: Tags words outside any keyphrase.

  - These tags were converted to numerical values for the model.

4.3. Oversampling:

To address class imbalance (common in keyphrase datasets), examples containing `B` or `I` tags were oversampled.This ensured sufficient representation of keyphrases, leading to improved learning during model training.

4.4. Embedding and Tokenization:

- SciBERT Tokenizer:

Tokenizes text while preserving the alignment with BIO tags, handling sub word tokenization (e.g., splitting "keyphrase" into "key" and "phrase"). Maintains token-label correspondence even for split tokens.

- Dynamic Embeddings:

Each token embedding is contextually enriched, capturing semantic relationships within the sentence.

4.5. Model Fine Tuning :

- SciBERT Architecture:

SciBERT is fine-tuned for token classification with three labels (`B`, `I`, and `O`).Pre-trained weights were partially frozen and gradually unfrozen during training to balance generalization and task-specific learning.

- Gradual Unfreezing:

  - Initial Training (Phase 1):

- Upper layers were trainable while lower layers were frozen to retain pre-trained knowledge.

- Three epochs were conducted to refine task-specific layers.

- Advanced Training (Phase 2):

  - Additional layers were unfrozen, allowing deeper fine-tuning of the model.

  - Five more epochs ensured robust learning for scientific text.

- Hyperparameters:

- Learning Rate: `2e-5`.

- Batch Size: 8.

- Weight Decay: 0.01 for regularization.

- Evaluation: Metrics calculated at the end of each epoch.

4.6. Evaluation:

- Model predictions were compared to ground truth labels using metrics like precision, recall, F1-score, and accuracy.The sequence tagging nature of the task required alignment between tokenized words and BIO tags, ensuring meaningful evaluations.

4.7. Embedding Effectiveness

4.7.1. Dynamic Contextual Understanding:

Unlike static embeddings (e.g., Word2Vec, GloVe), SciBERT generates dynamic embeddings tailored to the context in which words appear. For example: The word "network" in "neural network" is semantically distinct from its use in "social network," and SciBERT captures this distinction.

4.7.2. Handling Absent Keyphrases:

- SciBERT can generate keyphrases that are semantically related to the document even if they are not explicitly mentioned.This ability is particularly useful in scientific texts where important concepts may not be directly stated.

4.7.3. Domain-Specific Adaptability:

- SciBERT, pre-trained on scientific corpora, excels in understanding technical jargon and scientific terminology.Compared to general-purpose models like BERT, SciBERT offers better alignment with scientific datasets like KP20k.

---

5. Performance Metrics:

- Accuracy: 99.08% — High alignment between predicted and true BIO tags.

- Precision: 84.45% — Proportion of correctly identified keyphrases among all predictions.

- Recall: 90.32% — Proportion of actual keyphrases correctly retrieved.

- F1 Score: 87.28% — A balanced metric considering both precision and recall.

---

6. Traditional Approaches:

The traditional approach implemented in the code uses **TF-IDF** (Term Frequency-Inverse Document Frequency) combined with a **Logistic Regression** classifier. The steps are as follows:

6.1 Data Preprocessing:

- Texts are cleaned by removing non-alphanumeric characters and converting them to lowercase.
- Stop words (common words with little significance) are removed to focus on relevant terms.

6.2 Feature Representation:

- TF-IDF is used to convert the cleaned text into numerical vectors based on term frequencies adjusted for their rarity across the dataset.
- The maximum number of features (words) is set to 5000.

6.3 BIO Tagging:

- The labels ("B", "I", "O") are converted into numeric values (e.g., 0 for "O").
- Each document is assigned a flattened label based on its most frequent tag.

6.4 Model Training and Evaluation:

- The dataset is split into 80% training and 20% testing.
- A Logistic Regression model is trained to classify the labels using the TF-IDF features.
- The model is evaluated using metrics such as Accuracy, Precision, Recall, and F1-Score.

6.5 Results of the Traditional Approach

The traditional approach's performance is summarized below:

- **Accuracy:** 86.4%
- **Precision:**
  - "O" (Outside): 86%
  - "B" (Beginning): 0% (no correct predictions)

- **Recall:**
    - "O": 100%
    - "B": 0%
- **F1-Score:**
    - "O": 93%
    - "B": 0%
- **Weighted Averages:**
    - Precision: 75%
    - Recall: 86%
    - F1-Score: 80%

These results demonstrate reasonable performance for identifying non-keyphrase text but significant challenges in predicting keyphrase boundaries ("B").

---

7. Comparison Between Traditional and Advanced Methods

Metric	Traditional (TF-IDF)	Advanced (BERT)
Accuracy	86.4%	99.08%
Precision	75%	84.45%
Recall	86%	90.32%
F1-Score	80%	87.28%

7.1 Key Differences:

7.1.1 Contextual Understanding:

- **Traditional:** Relies solely on word frequencies, which limits its ability to understand word meanings in context.
- **BERT:** Leverages embeddings that capture bidirectional contextual information, allowing for better differentiation between similar terms in different contexts.

7.1.2 Handling of Keyphrase Boundaries:

- **Traditional:** Fails to identify "B" tags effectively due to the lack of sequence-awareness.
- **BERT:** Uses sequence tagging, which excels at recognizing the start and continuation of keyphrases.

7.1.3 Absent Keyphrases:

- **Traditional:** Cannot handle keyphrases that are semantically implied but not explicitly stated.
- **BERT:** Excels in generating semantically relevant absent keyphrases.

Aspect	Traditional Approaches	SciBERT (Embedding-Based)
Context Awareness	Relies on co-occurrence	Captures bidirectional context
Absent Keyphrases	Not supported	Fully supported
Domain Adaptability	Limited	High (scientific focus)
Semantic Richness	Surface-level features	Deep semantic understanding

---

7.4 Weaknesses of the Traditional Approach

1. **Limited Context Awareness:** TF-IDF only accounts for word frequency and ignores the surrounding words, leading to suboptimal performance in identifying phrases.
2. **Poor Boundary Detection:** The method struggles with sequence tagging, failing to correctly predict where keyphrases begin and end.
3. **Absent Keyphrases:** Traditional methods are inherently incapable of generating keyphrases that are not explicitly present in the text.
4. **Bias Toward Frequent Terms:** High-frequency words dominate predictions, reducing the diversity of extracted keyphrases.

---

8. Analysis of Results

The advanced method using BERT embeddings significantly outperforms the traditional TF-IDF-based method. This superiority is attributed to:

- **Dynamic Embeddings:** BERT embeddings are context-sensitive, allowing the model to differentiate between words with multiple meanings.

- **Sequence Tagging:** BERT's token classification capabilities enable it to predict BIO tags more accurately, even for rare cases.
- **Domain Adaptability:** SciBERT (a domain-specific BERT variant) enhances performance on scientific texts like KP20k.

---

9. Practical Benefits:

- Improved Keyphrase Selection:
  - Context-aware embeddings enable better identification of relevant keyphrases.
- Robustness Across Domains:
  - The model can generalize to other technical domains with minimal fine-tuning.
- Scalability:
  - Embedding-based methods can scale effectively to large datasets.

---

10. Conclusion

The combination of dynamic embeddings from SciBERT and a robust training methodology (gradual unfreezing, oversampling, etc.) significantly enhances keyphrase extraction performance. This embedding-based approach not only outperforms traditional frequency-based methods but also demonstrates superior adaptability and semantic understanding, making it a powerful tool for NLP tasks in specialized domains like scientific literature.

---

11. Insight

The comparison between traditional and advanced methods for keyphrase extraction reveals a paradigm shift in natural language processing. While traditional approaches like TF-IDF are computationally efficient and simple to implement, they fall short in handling the nuanced challenges of keyphrase extraction, such as contextual understanding, boundary detection, and absent keyphrases.

On the other hand, BERT-based models, particularly domain-specific variants like SciBERT, demonstrate a significant leap in performance by leveraging contextual embeddings and sequence tagging techniques. This highlights the importance of adopting advanced models for complex tasks in specialized domains, as they provide both higher accuracy and deeper semantic understanding. The results underscore the need for modern NLP systems to move beyond frequency-based techniques and embrace embedding-based methodologies to meet the demands of real-world applications.

12.References

1. Meng, R., Zhao, S., Han, S., et al. (2017). Deep Keyphrase Generation.Available at (https://arxiv.org/abs/1704.06879).

2. Zhang, Q., Wang, Y., Gong, Y., & Huang, X. (2016). Keyphrase Extraction Using Deep Recurrent Neural Networks on Twitter.* Proceedings of the 2016 EMNLP.

3. KP20k Dataset. Available at [Dataset Repository] (https://example.com/kp20k).